



Proyecto Final de Carrera

Sistema de Gestión de Contenidos para plataforma Aptshop

Autor: Daniel Peña Gil
Tutor: Álvaro Luis Bustamante

I Agradecimientos

Quisiera agradecer a mis compañeros de clase Alberto, Federico y Daniel y de residencia Juan, Raúl, Luis y Álvaro el hecho de que hicieran mi paso por la Universidad una de las mejores experiencias de mi vida. A mis compañeros de prácticas Enrique y Jorge, por inspirarme y ser unos ejemplos a seguir. A Bonifacio, porque sin él, hoy no estaría donde estoy. A Clara, por sus buenos consejos. A Irene, que tanto me ha apoyado en este último tramo de carrera. A Jorge y a Ana, que son mis incondicionales. Pero sobre todo, me gustaría agradecer a mis padres, Julia y Miguel Ángel. Sin su apoyo, dedicación e insistencia, hoy no estaría presentando este proyecto.

II Resumen

Los sistemas de gestión de contenidos son la solución más utilizada para crear páginas web y el software de comercio electrónico ha ganado mucha popularidad recientemente. Este proyecto examina la viabilidad de utilizar la combinación de ambos en el mismo proyecto y describe el análisis y desarrollo de un sistema de gestión de contenidos para la plataforma de comercio electrónico Aptshop.

III Abstract

Content management systems are the most widely used solution for creating web pages and e-commerce software has gained a lot of popularity recently. This project examines the feasibility of using the combination of both in the same project and describes the analysis and development of a content management system for the Aptshop e-commerce platform.

IV Tabla de contenido

I	Agradecimientos.....	iii
II	Resumen.....	iv
III	Abstract.....	v
IV	Tabla de contenido.....	vi
V	Índice de ilustraciones	x
VI	Índice de tablas.....	xii
1	Introducción	15
1.1	Contexto	15
1.2	Motivación.....	18
1.3	Objetivos.....	19
1.4	Alcance	19
1.5	Estructura del documento	20
2	Estado del arte.....	21
2.1	Gestores de contenido	21
2.2	Tipos de gestores de contenido	21
2.2.1	Blog: Wordpress	22
2.2.2	CMS: Joomla!	24
2.2.3	CMS: Drupal.....	27
2.2.4	eCommerce: Magento	29
2.2.5	eCommerce: PrestaShop	31
2.2.6	eCommerce: Aptshop.....	35
2.3	Elección de software.....	38
3	Análisis del sistema	40
3.1	Definición del sistema: Alcance, restricciones y entorno operacional....	40
3.1.1	Alcance	40
3.1.2	Restricciones	40
3.1.3	Entorno operacional	40
3.2	Especificación de requisitos de usuario	41
3.2.1	Requisitos de usuario funcionales.....	42
3.2.2	Requisitos de usuario no funcionales.....	50
3.3	Casos de Uso.....	57
3.4	Especificación de Requisitos de Software	76

3.4.1	Requisitos de software funcionales.....	77
3.4.2	Requisitos de software no funcionales.....	88
3.1	Matriz de trazabilidad de requisitos.....	96
4	Diseño.....	97
4.1	Arquitectura Aptshop	97
4.1.1	Arquitectura Front-office	97
4.1.2	Arquitectura Back-office.....	102
4.2	Vista general	103
4.3	Front-office.....	103
4.3.1	Modelo	105
4.3.2	Controlador	107
4.3.3	Vista	112
4.4	Back-office.....	113
4.4.1	Activación de modo edición	113
4.4.2	Pantallas de gestión.....	113
5	Construcción de la solución	117
5.1	Acceso al modo edición	117
5.2	Front-office en modo edición.....	117
5.3	Back-office, gestión de página de CMS	119
5.4	Módulo de <i>banners</i>	121
5.5	Módulo de producto	122
5.6	Módulo de testimonios de clientes.....	123
5.7	Módulo de vídeo.....	125
5.8	Módulo de listado de productos	125
5.9	Módulo de título y descripción	127
5.10	Estructura de columnas	128
5.11	Estructura de puntos cardinales	129
5.12	Estructura de pestañas.....	130
5.13	Estructura de pase de diapositivas.....	131
5.14	Decoradores	131
6	Gestión del proyecto	135
6.1	Introducción.....	135
6.2	Estimación de recursos	135
6.2.1	Recursos hardware	135
6.2.2	Recursos software	135

6.2.3	Recursos humanos	135
6.3	Plantificación	136
6.3.1	Diagrama de Gantt	137
6.4	Presupuesto	138
6.4.1	Costes directos	138
6.4.2	Costes indirectos	138
6.4.3	Costes totales	139
7	Conclusiones y trabajos futuros	140
7.1	Resultados obtenidos	140
7.1.1	Estandarización en el desarrollo de front-office	140
7.1.2	Mejoras en la calidad de servicio	140
7.1.3	Mejoras en tiempos de desarrollo	140
7.1.4	Eficiencia en proyectos actuales	140
7.1.5	Nuevos proyectos	141
7.1.6	Retorno de la inversión	141
7.2	Dificultades encontradas.....	141
7.3	Líneas futuras	141
8	Bibliografía	143
9	Anexo I: JSON Schema de los módulos de CMS.	145
9.1	Módulo <i>Banners</i>.....	145
9.2	Módulo Producto.....	145
9.3	Módulo Testimonio Clientes.....	145
9.4	Módulo Imágenes	146
9.5	Módulo Vídeo.....	146
9.6	Módulo Listado de Productos	146
9.7	Módulo Plantillas Predefinidas	147
9.8	Módulo Formulario Captación	147
9.9	Módulo Título y Descripción	148
9.10	Estructura de Columnas	150
9.11	Estructura de Puntos Cardinales.....	151
9.12	Estructura de Pestañas	151
9.13	Estructura de Diapositivas	151
9.14	Decorador de Título	151
9.15	Decorador de Enlace	152
9.16	Decorador de Bloque.....	152

10	Anexo II: JSON Schema de las páginas de CMS.....	153
-----------	---	------------

V Índice de ilustraciones

Ilustración 1. Logotipo Wordpress	22
Ilustración 2. Panel de gestión de WordPress.....	24
Ilustración 3. Logotipo Joomla!	24
Ilustración 4. Panel de gestión de Joomla!.....	27
Ilustración 5. Logotipo Drupal.....	27
Ilustración 6. Panel de gestión de Drupal	29
Ilustración 7. Logotipo Magento.....	29
Ilustración 8. Panel de gestión de Magento	31
Ilustración 9. Logotipo PrestaShop.....	31
Ilustración 10. Panel de gestión de PrestaShop	33
Ilustración 11. Uso de gestores de contenido en la web	34
Ilustración 12. Logotipo Aptitude / Aptshop	35
Ilustración 13. Panel de gestión de Aptshop.....	38
Ilustración 14. Estructura de puntos cardinales	44
Ilustración 15. Casos de uso.....	58
Ilustración 16. Diagrama de clases front-office.....	98
Ilustración 17. Diagrama de secuencia front-office	101
Ilustración 18. Botones de edición CMS en front-office	105
Ilustración 19. Modelo relacional CMS.....	106
Ilustración 20. Gestión de URLs de CMS	108
Ilustración 21. Diagrama de secuencia de la Gestión de URLs de CMS.....	109
Ilustración 22. Controladores de vistas de CMS.....	111
Ilustración 23. JSON Editor por Jeremy Dorn.....	114
Ilustración 24. Acceso al modo edición desde el back-office.....	117
Ilustración 25. Front-office en modo edición	118
Ilustración 26. Insertar nuevo módulo.	119
Ilustración 27. Pantalla de edición de página.....	120
Ilustración 28. Módulo de banners (front-office)	121
Ilustración 29. Módulo de banners (back-office).....	122
Ilustración 30. Módulo de producto (front-office).....	123
Ilustración 31. Módulo de producto (back-office).....	123
Ilustración 32. Testimonios de clientes (front-office).....	124
Ilustración 33. Testimonios de clientes (back-office).....	124

Ilustración 34. Módulo de vídeo (front-office).....	125
Ilustración 35. Módulo de vídeo (back-office)	125
Ilustración 36. Módulo de listado de productos (front-office).....	126
Ilustración 37. Módulo de listado de productos (back-office).....	126
Ilustración 38. Módulo de título y descripción (front-office).....	127
Ilustración 39. Módulo de título y descripción (back-office)	128
Ilustración 40. Estructura de columnas (front-office).....	128
Ilustración 41. Estructura de columnas (back-office)	129
Ilustración 42. Estructura de punto cardinales (front-office).....	129
Ilustración 43. Estructura de puntos cardinales (back-office)	130
Ilustración 44. Estructura de pestañas (front-office).....	130
Ilustración 45. Estructura de pestañas (back-office).....	130
Ilustración 46. Estructura de pase de diapositivas (front-office)	131
Ilustración 47. Estructura de pase de diapositivas (back-office)	131
Ilustración 48. Gestión de decoradores.....	132
Ilustración 49. Gestión de decoradores. Decorador de título.....	132
Ilustración 50. Decorador de título (front-office)	133
Ilustración 51. Decorador de enlace (back-office).....	133
Ilustración 52. Decorador de bloque (back-office).....	134
Ilustración 53. Diagrama de Gantt	137

VI Índice de tablas

Tabla 1 - Requisito de ejemplo	42
Tabla 2. Requisito de Usuario RUF-001.....	43
Tabla 3. Requisito de Usuario RUF-002.....	43
Tabla 4. Requisito de Usuario RUF-003.....	44
Tabla 5. Requisito de Usuario RUF-004.....	45
Tabla 6. Requisito de Usuario RUF-005.....	45
Tabla 7. Requisito de Usuario RUF-006.....	46
Tabla 8. Requisito de Usuario RUF-007.....	46
Tabla 9. Requisito de Usuario RUF-008.....	47
Tabla 10. Requisito de Usuario RUF-009.....	47
Tabla 11. Requisito de Usuario RUF-010.....	47
Tabla 12. Requisito de Usuario RUF-011.....	48
Tabla 13. Requisito de Usuario RUF-012.....	48
Tabla 14. Requisito de Usuario RUF-013.....	48
Tabla 15. Requisito de Usuario RUF-014.....	49
Tabla 16. Requisito de Usuario RUF-015.....	49
Tabla 17. Requisito de Usuario RUF-016.....	49
Tabla 18. Requisito de Usuario RUF-016.....	50
Tabla 19. Requisito de Usuario RUNF-001.....	50
Tabla 20. Requisito de Usuario RUNF-002.....	50
Tabla 21. Requisito de Usuario RUNF-003.....	51
Tabla 22. Requisito de Usuario RUNF-004.....	51
Tabla 23. Requisito de Usuario RUNF-005.....	52
Tabla 24. Requisito de Usuario RUNF-006.....	52
Tabla 25. Requisito de Usuario RUNF-007.....	53
Tabla 26. Requisito de Usuario RUNF-008.....	53
Tabla 27. Requisito de Usuario RUNF-009.....	54
Tabla 28. Requisito de Usuario RUNF-010.....	54
Tabla 29. Requisito de Usuario RUNF-011.....	55
Tabla 30. Requisito de Usuario RUNF-012.....	55
Tabla 31. Requisito de Usuario RUNF-013.....	56
Tabla 32. Requisito de Usuario RUNF-014.....	57

Tabla 33. Estructura de tabla para casos de uso.....	59
Tabla 34. Caso de uso CU-001	59
Tabla 35. Caso de uso CU-002	60
Tabla 36. Caso de uso CU-003	61
Tabla 37. Caso de uso CU-004	62
Tabla 38. Caso de uso CU-005	63
Tabla 39. Caso de uso CU-006	64
Tabla 40. Caso de uso CU-007	65
Tabla 41. Caso de uso CU-008	66
Tabla 42. Caso de uso CU-009	67
Tabla 43. Caso de uso CU-010	68
Tabla 44. Caso de uso CU-011	69
Tabla 45. Caso de uso CU-012	70
Tabla 46. Caso de uso CU-013	71
Tabla 47. Caso de uso CU-014	72
Tabla 48. Caso de uso CU-015	72
Tabla 49. Caso de uso CU-016	73
Tabla 50. Caso de uso CU-017	74
Tabla 51. Caso de uso CU-018	75
Tabla 52. Requisito de Software RSF-001	77
Tabla 53. Requisito de Software RSF-002	78
Tabla 54. Requisito de Software RSF-003	79
Tabla 55. Requisito de Software RSF-004	80
Tabla 56. Requisito de Software RSF-005	81
Tabla 57. Requisito de Software RSF-006	81
Tabla 58. Requisito de Software RSF-007	82
Tabla 59. Requisito de Software RSF-008	82
Tabla 60. Requisito de Software RSF-009	83
Tabla 61. Requisito de Software RSF-010	84
Tabla 62. Requisito de Software RSF-011	84
Tabla 63. Requisito de Software RSF-012	85
Tabla 64. Requisito de Software RSF-013	85
Tabla 65. Requisito de Software RSF-014	86
Tabla 66. Requisito de Software RSF-015	86

Tabla 67. Requisito de Software RSF-016	87
Tabla 68. Requisito de Software RSF-017	87
Tabla 69. Requisito de Software RSF-018	88
Tabla 70. Requisito de Software RSNF-001.....	88
Tabla 71 Requisito de Software RSNF-002.....	89
Tabla 72 Requisito de Software RSNF-003.....	89
Tabla 73 Requisito de Software RSNF-004.....	90
Tabla 74 Requisito de Software RSNF-005.....	90
Tabla 75 Requisito de Software RSNF-006.....	91
Tabla 76 Requisito de Software RSNF-007.....	91
Tabla 77 Requisito de Software RSNF-008.....	92
Tabla 78 Requisito de Software RSNF-009.....	92
Tabla 79 Requisito de Software RSNF-010.....	93
Tabla 80 Requisito de Software RSNF-011.....	93
Tabla 81 Requisito de Software RSNF-012.....	94
Tabla 82. Requisito de Software RSNF-013.....	95
Tabla 83. Requisito de Software RSNF-014.....	95
Tabla 84. Matriz de trazabilidad de requisitos.....	96
Tabla 85. Planificación de tareas	137
Tabla 86. Costes directos	138
Tabla 87. Costes indirectos.....	138
Tabla 88. Costes totales	139

1 Introducción

1.1 Contexto

Este proyecto, se enmarca dentro del actividad de la empresa Aptitude Web Services, S.L. y del desarrollo evolutivo continuado que ésta hace sobre su principal producto, Aptshop, un programa de comercio electrónico orientado a la Web especializado en la ejecución y medición de resultados de las campañas de marketing online.

Aptitude Web Services, S.L. es una pequeña empresa fundada en 2012 por dos exempleados de Icox Web Consulting Spain tras su cese de actividad:

Fernando Serrano Elvira, Técnico en Empresas y Actividades Turísticas, comenzó su relación con el mundo de la informática en 1995 en Turespaña pasando posteriormente a formar parte del equipo de Lared S.L., empresa pionera en la introducción y utilización del sistema operativo Linux en España. Estuvo destinado en el Ministerio de Economía y Hacienda, Consejo General del Notariado, Muface, Multauto y Legálitas realizando principalmente funciones de Dirección Técnica y consultoría en el ámbito de la seguridad informática y comunicaciones. Posteriormente se incorporó a Nexotel S.L. como Project Manager en el ámbito de sistemas de comunicaciones de voz, centralitas y sistemas auxiliares. Después de tres años se incorpora a Securwap S.L. donde realiza, destinado en el Ministerio del Interior, funciones de dirección técnica en el ámbito de seguridad y defensa, desarrollando sistemas de monitorización y control. Por último se incorpora a Icox Web Consulting donde realiza funciones de Project Manager hasta el cese de actividad de la empresa.

Y Daniel Peña Gil, estudiante de Ingeniería Informática por la Universidad Carlos III de Madrid. Comienza a trabajar en una beca del grupo Tecnodoc, del departamento de Biblioteconomía y Documentación de la Universidad Carlos III de Madrid, bajo la tutela de Bonifacio Martín Galán y en un momento en el que las diplomaturas en Biblioteconomía y Documentación de la UC3M están dando el salto hacia un modelo semipresencial. En esta beca, desarrolla varios proyectos con Joomla entre los que destaca la web de Seguridad del Paciente, dependiente del entonces Ministerio de Sanidad y Consumo, ahora Ministerio de Sanidad, Servicios Sociales e Igualdad. Posteriormente fuera ya del departamento, colabora con el desarrollo y

evolución de otros sitios web mediante el uso, entre otros, de los siguientes programas de CMS¹:

- Joomla
- Drupal
- Wordpress
- osCommerce
- Moodle

La actividad principal de Icox Web Consulting era el desarrollo de software con tecnologías LAMP² basadas en programación PHP, base de datos MySQL, servidor Apache y sistemas Linux alojados en los servicios en la nube de Amazon Web Services. De la antigua empresa se heredaron varios proyectos entre los que destaca Aprendum Compra Colectiva, una *startup* orientada a la venta de cursos online.

La plataforma de Aprendum se ejecutaba sobre osCommerce, un software libre de comercio electrónico muy popular a principios de los 2000 y que fue la base sobre la que funcionó hasta que fue completamente sustituido por **Aptshop**, software desarrollado por Aptitude para dicho proyecto.

Al tratarse de un proyecto heredado, existía la doble necesidad de, por un lado, conservar la funcionalidad antigua y por otro, evolucionar el sistema para suplir las necesidades que iban surgiendo.

osCommerce está desarrollado mayormente mediante programación estructurada, no tiene soporte para extender mediante módulos ni *plugins* genéricos más allá de los de pago o envío. Debido a esto, extender la funcionalidad hizo que se perdiera la posibilidad de instalar fácilmente las actualizaciones, siendo ésta una cuestión primordial, sobre todo al tratarse de un software libre muy utilizado. No actualizar un software de estas características supone estar expuesto a los *exploits* arreglados en las actualizaciones.

¹ Content Management System por sus siglas en inglés o Sistema Gestor de Contenidos en castellano, es un programa informático que permite dar de alta, modificar y borrar los contenidos de una página o sitio web.

² LAMP es el acrónimo de Linux Apache MySQL y PHP, es un conjunto de sistemas formado por estas tecnologías, muy común en el desarrollo de páginas web dinámicas o aplicaciones web.

El 30 de mayo de 2012, con la mente puesta en una actualización del diseño de la página, se toma la decisión de crear Aptshop ya con arquitectura MVC, con orientación a objetos y extensible mediante *plugins* y módulos.

Aptshop es un software desarrollado íntegramente por Aptitude Web Services que inicialmente clonaba la funcionalidad de osCommerce:

- Venta de productos
- Categorías de productos
- Área personal
- Carrito de la compra
- Pasarela de pago
- Buscador
- Etc.

Y según avanzaba la empresa y crecían las necesidades, se le fueron añadiendo distintas funcionalidades:

- Sistema de análisis web integrado, orientado al marketing online
 - o Gestión de campañas de marketing
 - Captación
 - Prospección
 - Cierre
 - Fidelización
 - o Seguimiento de usuario multidispositivo
 - o Cálculo de coste de adquisición del cliente (CAC) por canal
 - o Cálculo de valor del cliente en su vida (*lifetime value* - LTV) por canal
 - o Embudos de conversión
 - o Estudios de cohortes
 - o Camino del usuario
 - o Modelos de atribución
 - o Cálculo del retorno de la inversión (ROI)
 - o Test A/B
- Buscador avanzado
- Filtros de productos
- Herramienta de creación de *newsletters* e integración con distintas empresas de envío masivo de comunicaciones comerciales por email:

- Integración con Mailchimp
- Integración con Senderglobal
- Integración con EmailVision
- Sistema de análisis de *newsletters*

1.2 Motivación

A medida que Aptitude captaba más clientes y que estos iban creciendo, las necesidades de horas de desarrollo se multiplicaban. Algunos de los requerimientos correspondían con mantenimiento de la plataforma, otros con desarrollo de nuevas funcionalidades, pero muchos de ellos estaban relacionados con la gestión de los contenidos de la web.

La gestión del catálogo estaba resuelta. Productos y categorías, con sus títulos, descripciones, imágenes, etc. se podían crear, listar, modificar y eliminar. Pero las peticiones de los clientes siempre iban más allá.

Solicitudes típicas eran:

- Creación de nuevas páginas.
 - *Landings* o páginas de aterrizaje para la captación de nuevos clientes.
 - Páginas de información de la empresa
 - Páginas de información de producto aparte de la página de producto en sí.
 - Páginas de información de alguna campaña especial.
 - Etc.
- Creación de formularios de captación.
- Creación de formularios de satisfacción o encuestas.
- Inclusión de *banners* principales en las diferentes tipos de páginas (de categoría, de filtro, de producto, en la página principal, etc.)
- Inclusión de *banners* secundarios en página principal, página de categoría, página de filtro o página de producto.
- Inclusión y modificación de módulos basados en código HTML.
 - Imágenes llamativas que dirijan el tráfico al producto.
 - Módulos que fomenten el *upselling*³.
 - Módulos que fomenten la venta cruzada⁴ (o *cross-selling*).

³ Técnica mediante la cual se intenta vender un producto más caro con características superiores al que está visualizando el usuario en la pantalla.

- Mensajes de confianza a la hora de pagar (Pago seguro, garantía de compra, etc.).
- Inclusión de imágenes.
- Etc.

Todas estas tareas son generalmente resueltas por un Sistema Gestor de Contenido o CMS por sus siglas en inglés (Content Management System), y no es necesario un perfil de programador para hacer uso de estas herramientas. Teniendo un CMS integrado en el sistema podríamos dar un gran valor añadido a nuestros clientes si lo gestionan ellos o invertir un menor tiempo y de un perfil humano más económico en este tipo de tareas si lo gestionamos nosotros. En cualquiera de los dos casos mejora la competitividad de la empresa.

Es cierto que hay gran cantidad de soluciones de CMS, tanto libres como privativas, como veremos en el capítulo Estado del arte. Pero lo cierto es que al tratar de integrarlas dentro de un sistema heredado, a veces resulta imposible debido a la incompatibilidad de arquitecturas.

1.3 Objetivos

Viendo las oportunidades que se mencionan, se plantea como objetivo del proyecto desarrollar un sistema de gestión de contenidos para la plataforma Aptshop. El sistema debe ser usable por un usuario no técnico, sin conocimientos de lenguaje de programación. Los usuarios con conocimientos básicos podrán gestionar módulos que se podrán añadir a los huecos o posiciones destinadas para ello en las plantillas. Algunos de estos módulos permitirán alterar la estructura del documento. Usuarios con conocimientos más avanzados, podrán incluir directamente código HTML, CSS o JavaScript.

La arquitectura del sistema debe proveer a los desarrolladores de un marco en el que crear nuevos módulos de manera estandarizada y rápida.

1.4 Alcance

El proyecto abarcará desde la adecuación de Aptshop a una arquitectura que permita la ampliación de su funcionalidad, pasando por la construcción del sistema de gestión de contenidos, hasta la solución de problema encontrados, como por ejemplo un sistema de caché para solventar problemas de rendimiento.

⁴ Técnica mediante la cual se intenta aumentar el valor de la cesta incentivando la venta al usuario de productos relacionados con otros que ya esté dispuesto a comprar.

1.5 Estructura del documento

El documento se compone de siete partes. Comienza con la introducción del proyecto, donde se destaca la motivación del desarrollo del mismo. Posteriormente se ahonda en el estado del arte de los sistemas de gestión de contenidos en la actualidad y se explica la justificación por la que se decide desarrollar uno nuevo. A continuación comienza la estructura típica de un desarrollo de ingeniería de software, empezando por el análisis, seguido del diseño y terminando por la planificación. En el siguiente punto se describe de un modo visual el funcionamiento de la aplicación. Y para finalizar, en las conclusiones, se enumeran las ventajas que ha supuesto desarrollar y adoptar este sistema.

2 Estado del arte

El germen de este proyecto se remonta a 2012, y en ese momento, los sistemas de gestión de contenido se encuentran en un estado maduro. Los sistemas gestores de contenido que más éxito han tenido en Internet coinciden en dos puntos clave para este proyecto: son software libre y están desarrollados en PHP. En este apartado se va a realizar una revisión del estado del arte de las diferentes alternativas de sistemas de gestión de contenidos existentes en el mercado y, al final del capítulo, se seleccionará la tecnología base para el desarrollo de nuestro proyecto.

2.1 Gestores de contenido

Un sistema de gestión de contenido (SGC por sus siglas en castellano), más conocido como CMS por sus siglas en inglés, *es un programa informático que permite crear una estructura de soporte (framework) para la creación y administración de contenidos, principalmente en páginas web, por parte de los administradores, editores, participantes y demás usuarios* [1]. Explicado de una manera más sencilla y práctica, es un programa que sirve para gestionar textos, imágenes, vídeos, documentos y otros tipos de representación de la información en un sitio web, y la disposición de los mismos en las páginas. Los CMS cuentan con una interfaz que controla una o varias bases de datos donde se aloja el contenido del sitio web. Permite manejar el contenido y el diseño del sitio por separado, lo que permite cambiar el diseño en cualquier momento sin tener que dar formato de nuevo al contenido previamente publicado.

2.2 Tipos de gestores de contenido

Existen distintos tipos de CMS en función de la especialización de los contenidos que manejen. Los hay que manejan contenido genérico, otros gestionan blogs, comunidades o redes sociales, otros sistemas para la gestión de aprendizaje, otros tiendas online, etc. A continuación se exponen los máximos representantes de código abierto en cada una de las categorías.

2.2.1 Blog: Wordpress



Ilustración 1. Logotipo Wordpress

Con un 27,9% de los sitios de Internet servidos, un 58,9% si reducimos la base a los sitios manejados mediante gestores de contenido [2], Wordpress es el software líder en de cuota de mercado, el software que sirve un mayor número de sitios de Internet. Se publicó su primera versión el 27 de mayo de 2003 [3]. Inicialmente se desarrolló como un sistema de gestión de *blogs*, pero se han desarrollado gran cantidad de extensiones que permiten convertirlo en gestor de casi cualquier tipo de contenido.

La enorme comunidad Wordpress se beneficia de una gran selección de plantillas, *plugins* y *widgets*. Pero los críticos con el sistema no solo dicen que es bastante complicado, sino que además es confuso y raro de utilizar. Wordpress además acarrea los problemas que cualquier otro software de éxito conlleva, y es que es un objetivo prioritario para los *hackers*. Debido a ello, los administradores deben actualizar con frecuencia sus instalaciones. Estas actualizaciones son distribuidas rápidamente por los desarrolladores de Wordpress después del descubrimiento de cualquier incidente de seguridad, pero no son siempre compatibles con todos los *plugins* o extensiones, con lo que, las actualizaciones de software suelen acarrear un gran trabajo para los administradores del sistema, especialmente en proyectos grandes.

Debido a la facilidad de instalación, Wordpress es muy bueno para pequeños proyectos, especialmente para *blogs*, que fue el cometido para el que fue diseñado originalmente. Es muy sencillo para un bloguero poner a funcionar su sitio web con un diseño atractivo y colgar periódicamente nuevos *posts*. Esa facilidad va disminuyendo a medida que el proyecto se complica y se le incluye funcionalidad mediante *plugins* u otros añadidos.

Las ventajas principales, son:

- Una gran comunidad de usuarios y desarrolladores.

- Instalación en cinco minutos a través de un asistente, incluidos muchos servicios online que instalan Wordpress en un clic.
- Interfaz de usuario intuitiva (no la de administración).
- Integración sencilla con *plugins* y otras extensiones.

Las desventajas, no obstante no son despreciables:

- Para que funcione como un CMS es preciso instalar *plugins* o extensiones.
- Es habitual encontrar fallos de seguridad en los *plugins* y extensiones de terceros.
- Su estabilidad y rendimiento son limitados en sitios web de alto nivel de tráfico.
- Una alta frecuencia de actualizaciones conlleva una gran labor administrativa de los sistemas.

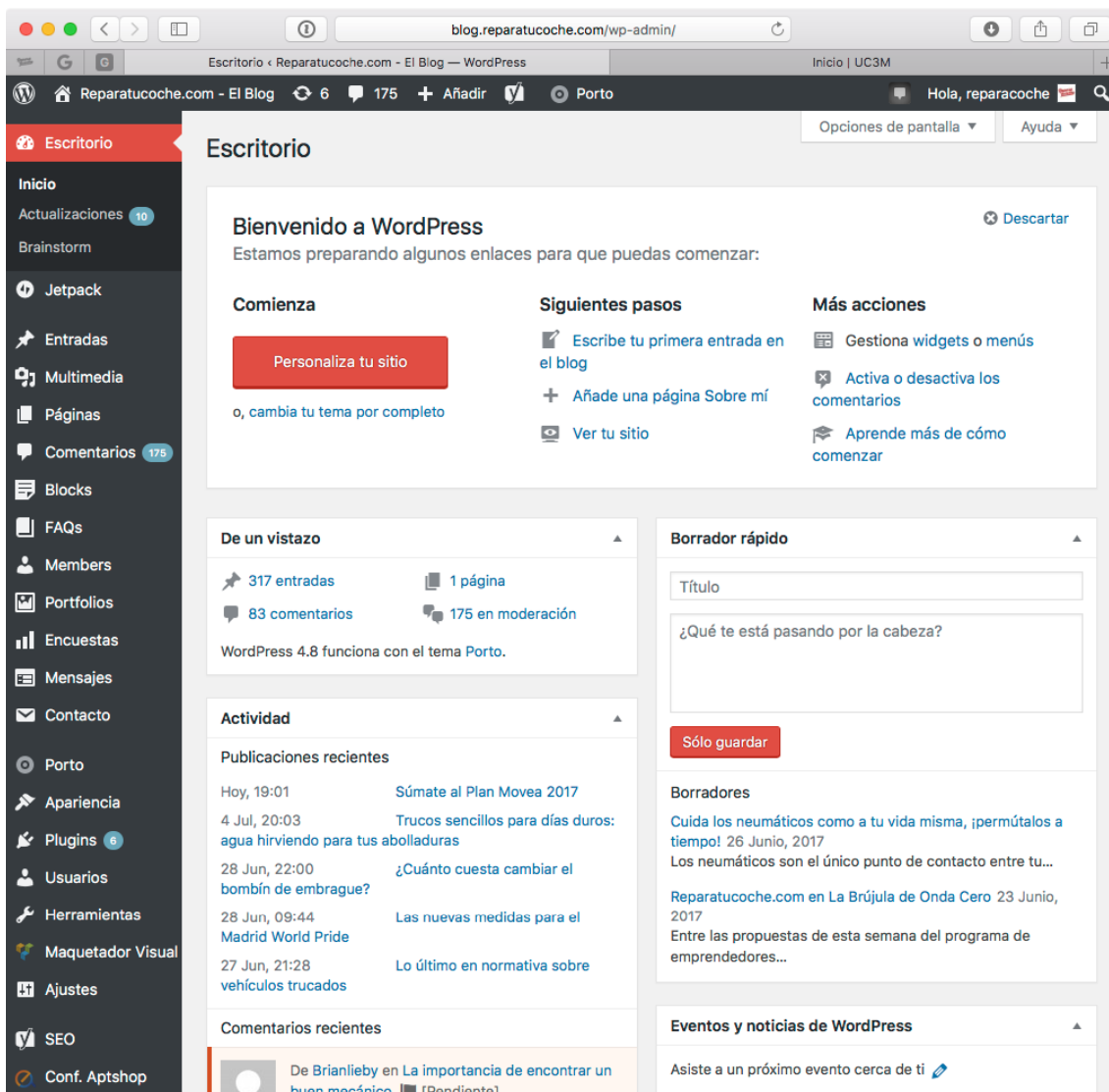


Ilustración 2. Panel de gestión de WordPress

2.2.2 CMS: Joomla!



Ilustración 3. Logotipo Joomla!

Con un 3,3% de los sitios de Internet servidos, un 7% si reducimos la base a los sitios manejados mediante gestores de contenido [2], Joomla! es el segundo CMS en cuota de mercado. Es especialmente utilizado en los EE.UU. donde supera por casi diez veces a Alemania, segundo país por instalaciones de Joomla! [4]. La primera versión de Joomla (1.0.0) se publicó el 16 de septiembre de 2005 y se trataba inicialmente de un *fork* o versión paralela de Mambo 4.5.2.3 combinada con mejoras de seguridad y corrección de fallos [5]. Mambo es otro CMS del que no trataremos en este trabajo debido a que su desarrollo se vio suspendido en 2008 y desde entonces no ha recibido soporte ni actualizaciones [6].

Joomla! se caracteriza, al contrario que Wordpress, por estar desarrollado, mediante el paradigma de programación orientación a objetos y basado además en el patrón de diseño MVC. Esto permite a usuarios de Joomla! escribir sus propias extensiones de una manera más sencilla. Hay cinco tipos de extensiones en Joomla! [7]:

- Componentes
- Módulos
- *Plugins*
- Plantillas
- Idiomas

Los componentes de Joomla son extensiones que están asociados a una página o URL del sitio. Siempre hay uno y sólo un componente asociado a una página. Declaran el contenido principal de la página y es el elemento que dota de identidad a la misma.

Los módulos, igual que los componentes, permiten añadir un contenido de un tipo específico a la página. Sin embargo, este contenido no es el principal de la misma, sino que se añade en las posiciones que se crean en las plantillas.

Los *plugins*, por otro lado, no están asociados a ninguna página o posición de plantilla concreta, sino que se pueden activar y desactivar para el sitio completo. Ejecutan código en base a eventos lanzados por el núcleo de Joomla! y los hay de distintos tipos:

- De usuario
- De búsqueda
- De contenido
- Etc.

Las plantillas forman parte de la vista del Sistema. Deben dar soporte a los componentes instalados, para que cada uno pueda verse representado con mediante el *look and feel* de la plantilla.

Y para finalizar, los idiomas son un conjunto de ficheros que almacenan los pares clave, valor de cada uno de los textos que utiliza la aplicación.

Los cinco tipos de extensiones pueden ser utilizados tanto para el *frontend* como para el *backend* del sitio, dando lugar en ocasiones a confusiones, malentendidos y complicaciones tanto a usuarios que desean utilizar dichas extensiones como a los desarrolladores.

Los desarrolladores pueden publicar sus extensiones en el directorio de extensiones Joomla! (Joomla! *Extensions Directory*), y los usuarios pueden instalar fácilmente estas extensiones desde el *backend* de su Joomla! La gestión de derechos sobre el software desarrollado y los procesos de aprobación para la inclusión de una extensión en el directorio han sido en ocasiones motivo de queja de los desarrolladores, debido a la dificultad de estos. Por otro lado, los usuarios ven como principal punto negativo el hecho de que algunas de las mejores extensiones sean de pago.

Las principales ventajas son:

- Instalación y configuración sencillas
- Aunque menor que Wordpress, gran cantidad de extensiones y diseños
- Creación de plantillas sencillas

Y las principales desventajas son:

- Gestión de derechos inadecuada
- Extensiones especializadas pueden llegar a ser caras.
- Las extensiones a veces requieren acciones manuales por parte de los administradores de sistema para su correcta instalación.

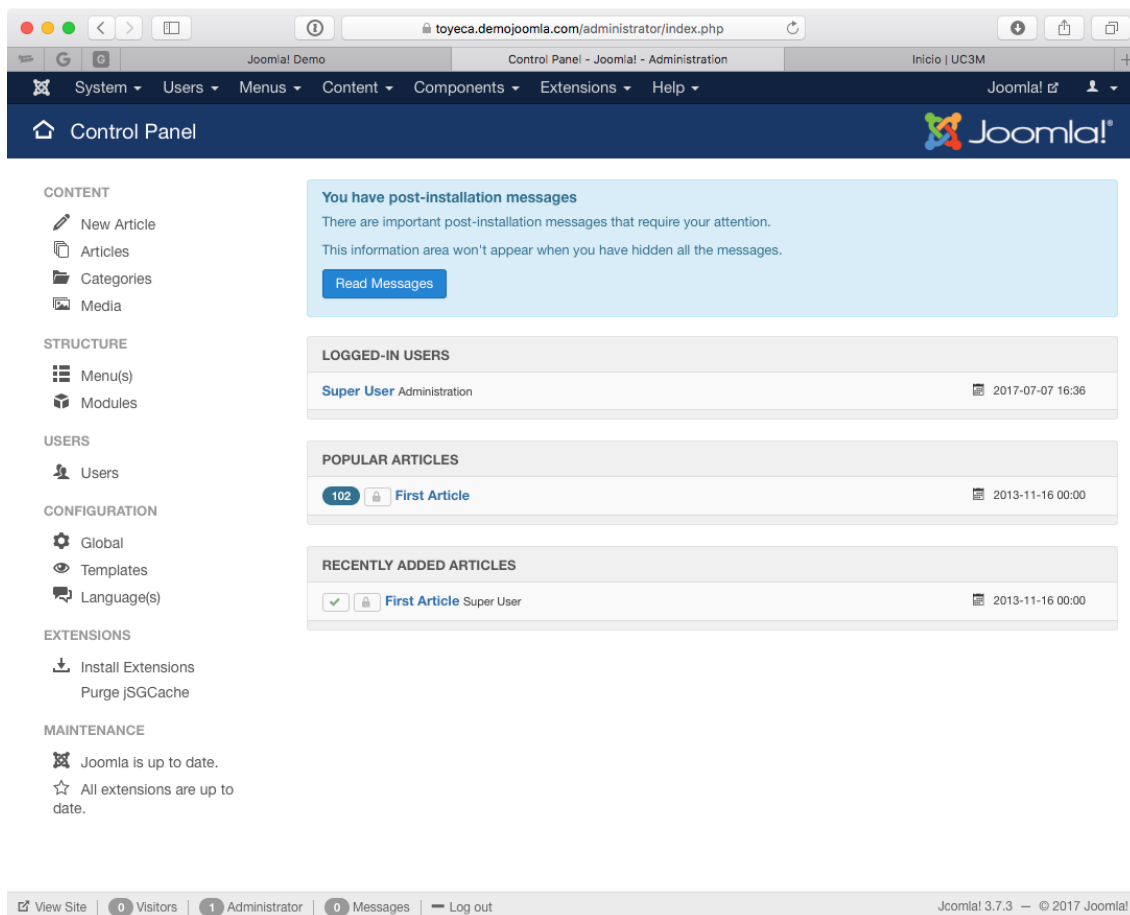


Ilustración 4. Panel de gestión de Joomla!

2.2.3 CMS: Drupal



Ilustración 5. Logotipo Drupal

Con un 2,2% de los sitios de Internet servidos, un 4,7% si reducimos la base a los sitios manejados mediante gestores de contenido [2], Drupal es el tercer CMS en de cuota de mercado. La primera versión de Drupal se publicó el 1 de enero de 2001, y surgió como una solución para comunidades estudiantiles, pero pronto se convirtió en un CMS de pleno derecho [8]. La instalación básica de Drupal tiene una funcionalidad muy limitada, siendo capaz de ejecutarse de una manera muy rápida, aunque puede ampliarse enormemente utilizando módulos. No obstante, a pesar de la ligereza, la

instalación básica cuenta, entre otros, con un sistema de *blogs*, foros y nubes de palabras clave.

Debido a su sistema modular, el CMS Drupal permite una gran personalización del gestor. Además, los usuarios que saben exactamente lo que necesitan, pueden descargarse versiones de Drupal preconfiguradas justo con las extensiones que se ajusten a sus necesidades.

Con el foco en publicaciones sociales, Drupal puede ser perfecto para proyectos de tamaño pequeño o mediano. Si queremos utilizarlo para grandes proyectos, es necesario un conocimiento y formación específicos.

Las principales ventajas son:

- Un software inicialmente liviano
- Una gran cantidad de módulos para ampliar la funcionalidad y distribuciones personalizadas de Drupal ya preconfiguradas.
- Manejo multidominio.

Y como desventajas:

- Configuración compleja debido a la pobre compatibilidad del *backend*.
- La principal ventaja se puede convertir en un inconveniente, el hecho de que el software sea inicialmente liviano conlleva que se tengan que instalar las ampliaciones por separado.
- La instalación de módulos sólo es posible mediante FTP.

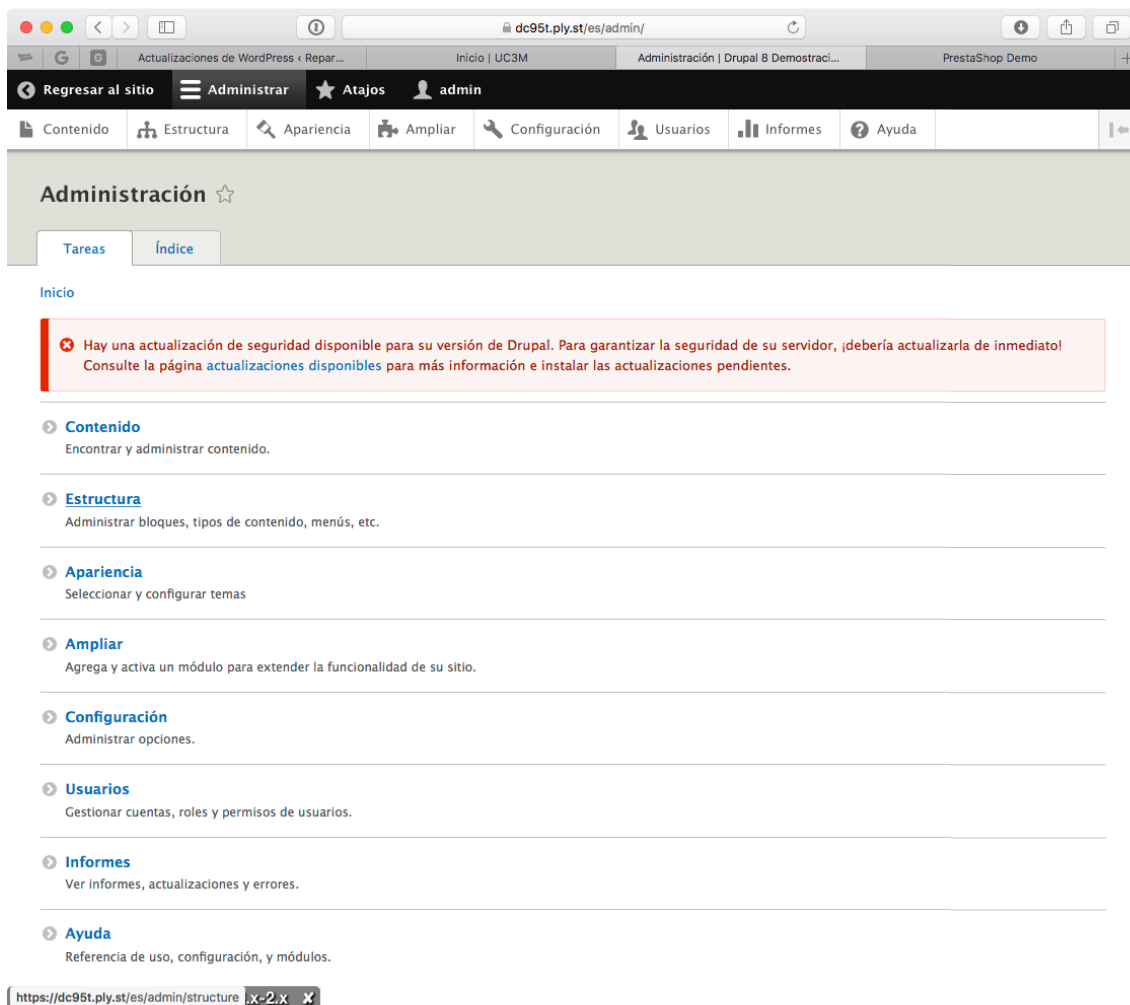


Ilustración 6. Panel de gestión de Drupal

2.2.4 eCommerce: Magento



Ilustración 7. Logotipo Magento

Magento ya entra dentro de un tipo de producto mucho más específico. Orientado a sitios de comercio electrónico, Magento es el software líder en CMS en este sector, habiendo sido instalado en un 1,2% de los sitios de Internet o en un 2,5% si reducimos la base a los sitios manejados mediante gestores de contenido [2].

Magento fue lanzado al público el 31 de marzo de 2008 por Varient Inc. [9] y posteriormente fue adquirida por eBay, creando dos versiones:

- Community Edition

- Enterprise Edition

La versión Enterprise Edition ofrece soporte técnico, integraciones con software de terceros como CRM (Sistemas de gestión de clientes) y ERP (Sistema de planificación de recursos empresariales), optimización en algunos procesos y módulos avanzados. El coste de la licencia es de 15.000€/año.

El núcleo de Magento está construido utilizando el marco de desarrollo Zend Framework, aunque lo cierto es que a los desarrolladores no debería influirles ya que Magento tiene su propio API de acceso a los recursos de la plataforma y se desaconseja la llamada directa al API Zend. Las extensiones de Magento se dividen en tres tipos de componentes:

Módulos: Interactúan con otras partes de la aplicación para gestionar o proveer de alguna funcionalidad de negocio. Pueden contener una interfaz de usuario para mostrar información o permitir la interacción con el usuario. Además pueden contener interfaces de aplicación con la posibilidad de ser llamadas desde otro módulo.

- Temas o plantillas: Proveen de vistas personalizadas que permiten modificar el *look and feel* del *front-end* de la tienda o del panel de administración.
- Paquetes de idiomas: Permiten la gestión de la internacionalización y localización del sistema, tanto de la tienda como del panel de administración, proveyendo de las cadenas de caracteres y configuraciones necesarias que para mostrar el contenido en el idioma correctamente.

Las principales ventajas de Magento orientadas al comercio electrónico son:

- Múltiples tiendas virtuales pueden compartir un sólo panel de administración.
- Incorpora un servicio web que permite a aplicaciones externas acceder a los datos de Magento.
- Orientado a SEO con URLs amigables y datos estructurados.

Y las desventajas:

- El entorno es flexible pero complejo, difícil de aprender.
- Requiere una gran cantidad de procesamiento del servidor. Esto puede impactar directamente en el rendimiento de la aplicación y este impacto se incrementa considerablemente con la

personalización del sistema Magento para adecuarse a las necesidades de la empresa.

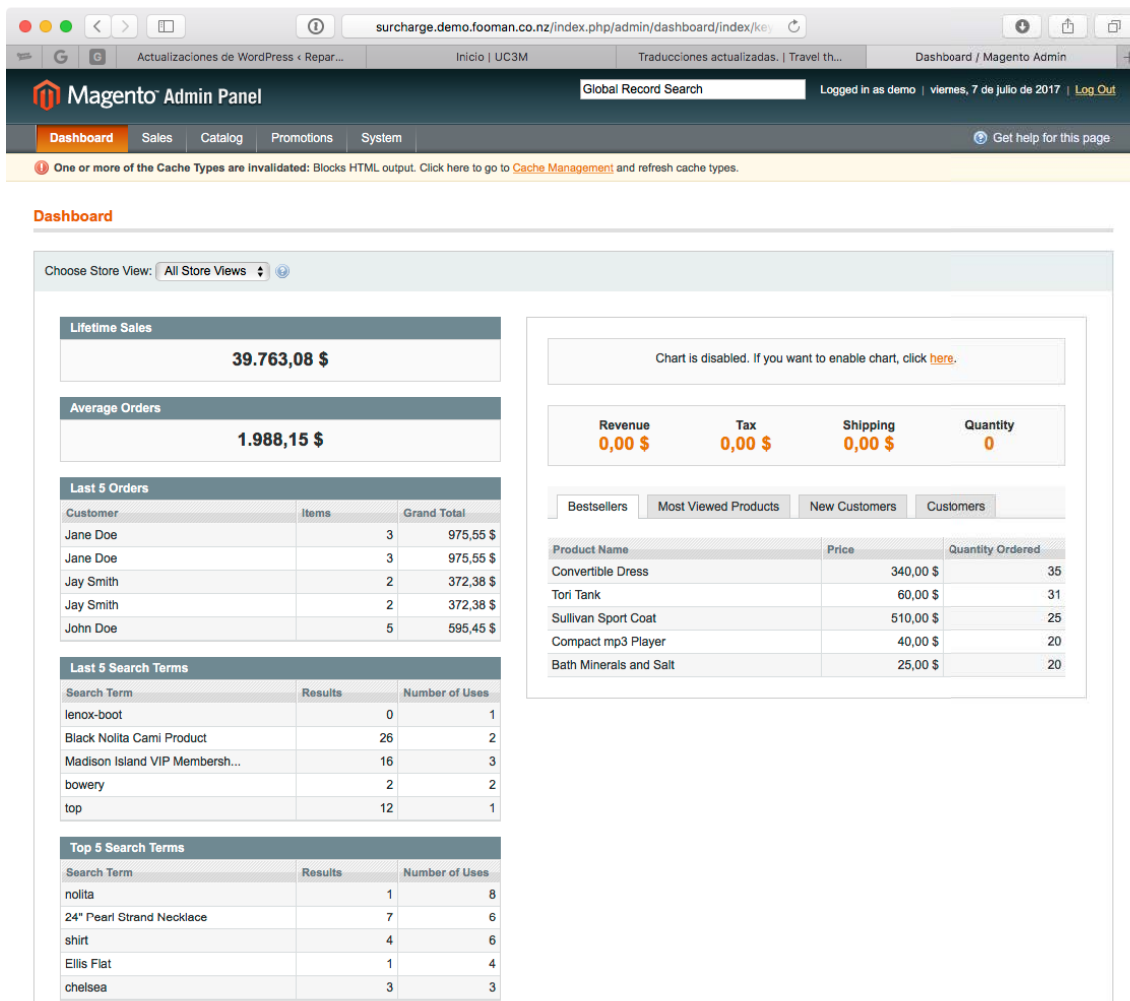


Ilustración 8. Panel de gestión de Magento

2.2.5 eCommerce: PrestaShop



Ilustración 9. Logotipo PrestaShop

PrestaShop, al igual que Magento es un producto orientado a sitios de comercio electrónico. PrestaShop es segundo en el ranking de soluciones eCommerce CMS libres

más utilizadas, sirviendo un 0,6% de los sitios de Internet o en un 1,3% si reducimos la base a los sitios manejados mediante gestores de contenido [2].

PrestaShop goza de especial popularidad en Europa. La primera versión fue lanzada el 31 de julio de 2005 como proyecto estudiantil de la EPITECH IT School de París. Aunque no fue hasta 2007 cuando Igor Schlumberger y Bruno Lévêque crearon la compañía PrestaShop S.A. ubicada también en París [10]. Debido a su modelo de desarrollo de software *open source*, tuvieron que resolver el reto de la monetización del modelo de negocio optando por dos soluciones:

- Venta de *addons* propios y de terceros en el *marketplace* de PrestaShop
- Alianzas estratégicas con líderes en el sector del comercio electrónico como PayPal o Google [11].

PrestaShop puede ampliarse mediante la creación de plantillas (*templates*) y módulos. El sistema de plantillas está desarrollado con Smarty, una de las soluciones más longevas y estables dentro del ecosistema PHP. Los módulos, son pequeñas aplicaciones que se integran en las tiendas PrestaShop, agregando funcionalidad tanto en la tienda como en el panel de administración.

Las principales ventajas de PrestaShop, desde el punto de vista de plataforma de comercio electrónico, son:

- El entorno es sencillo y es relativamente fácil implementar cambios en los módulos.
- No tiene unos grandes requerimientos de servidor.
- Gran comunidad de usuarios y multitud de plugins descargables.
- Mediante configuración se puede conseguir una buena optimización SEO

Y las desventajas:

- La administración es difícil con logística compleja o productos no estándar y existen problemas en catálogos grandes.
- No dispone de multi-tienda
- No dispone de servicio web de conectividad de forma nativa.
- Problemas de seguridad con las actualizaciones.

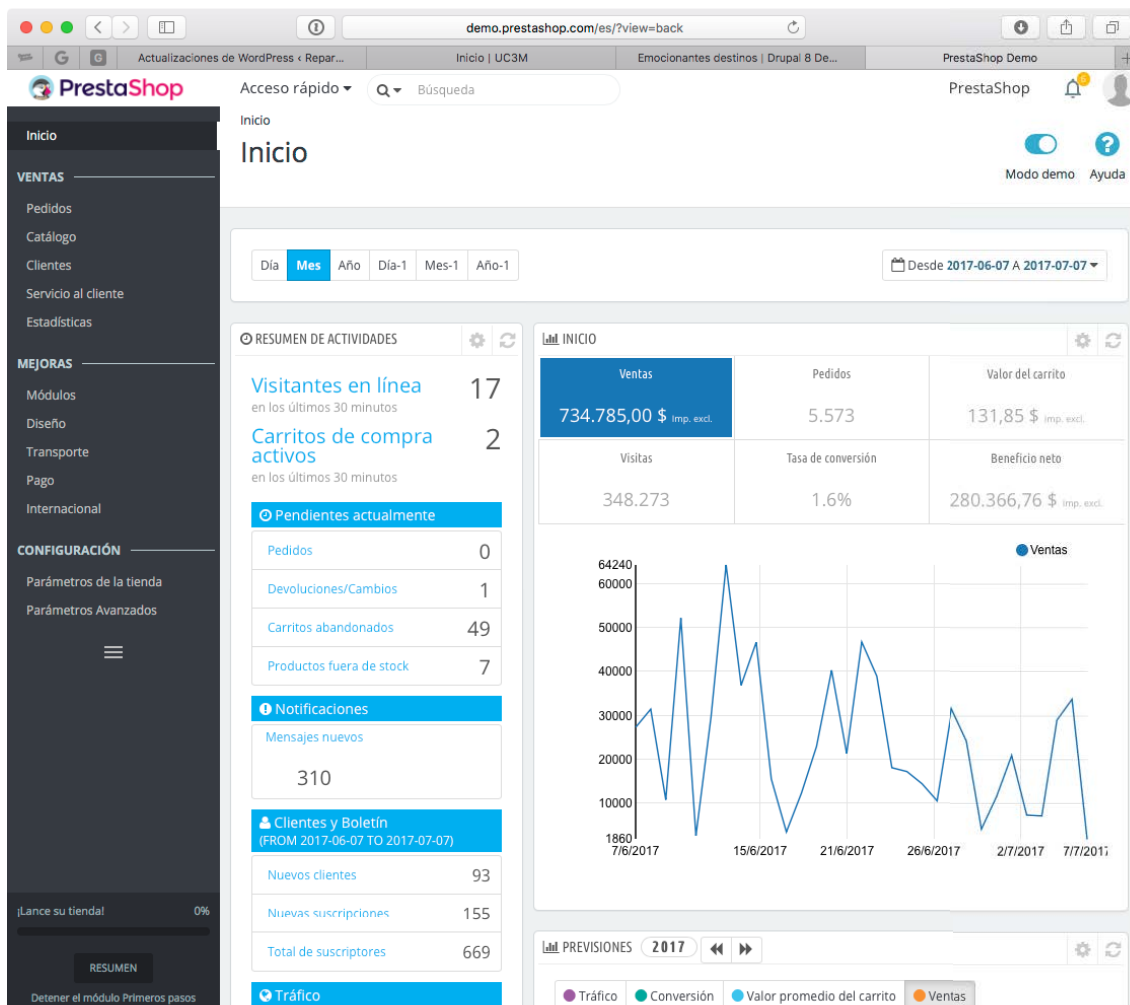


Ilustración 10. Panel de gestión de PrestaShop

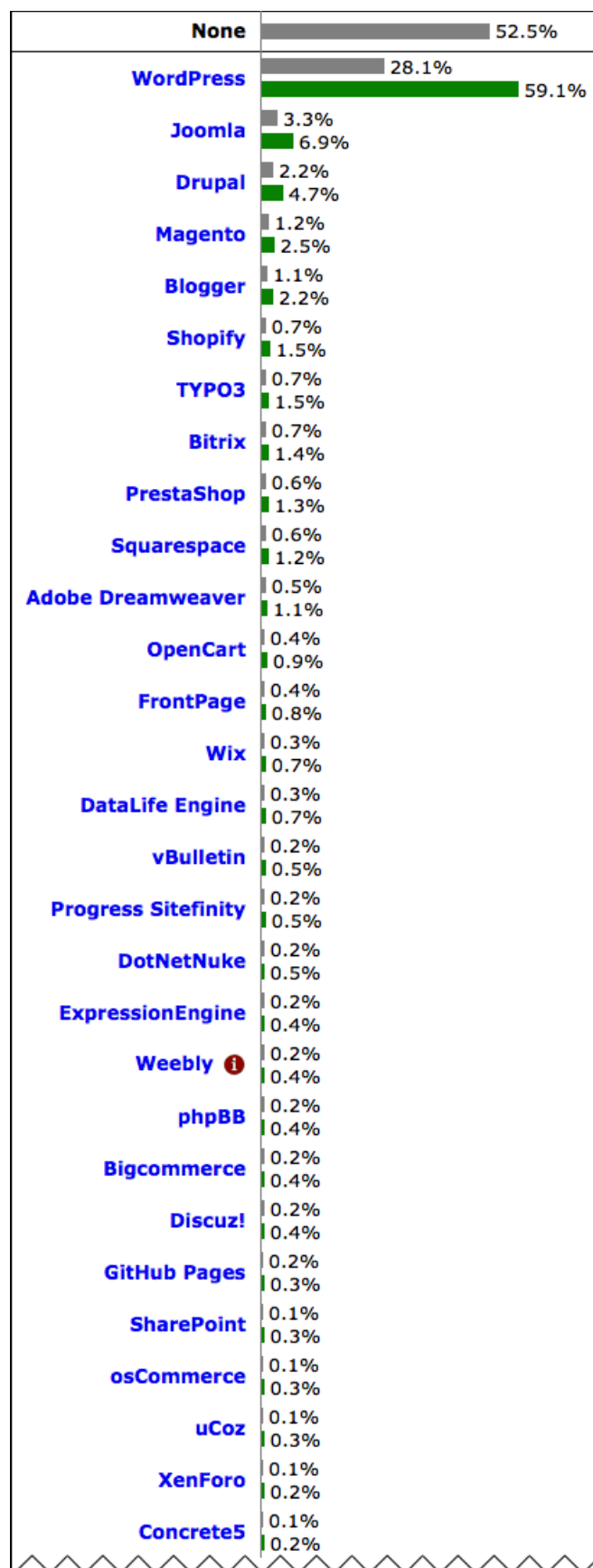


Ilustración 11. Uso de gestores de contenido en la web

2.2.6 eCommerce: Aptshop



Ilustración 12. Logotipo Aptitude / Aptshop

Aptshop, al contrario que los anteriores, no goza de su popularidad ni de un gran número de instalaciones, ya que es un producto propietario de una pequeña compañía. Es, básicamente, una plataforma de comercio electrónico a la que se han añadido importantes mejoras orientadas a la consecución de objetivos. En sus orígenes, la implementación de éstas mejoras surgió como una necesidad propia, no como un requerimiento de ningún cliente, ya que Aptitude formaba parte del accionariado de la *startup* Aprendum.com y era además responsable de todo su departamento técnico. Como en casi todas las *startups*, la financiación era el principal problema y era imprescindible que el software ofreciera el mayor número de herramientas posibles de cara a optimizar la rentabilidad del negocio. Aptshop se diferencia de las anteriores plataformas precisamente en eso, en su orientación al negocio, al análisis de la información, a proveer de datos que ayuden en la toma de decisiones y de herramientas al departamento de Marketing. Se planificaron y desarrollaron herramientas divididas en dos grandes grupos:

Sistema de análisis de tráfico: En la actualidad cualquier plataforma de comercio electrónico tiene a su disposición multitud de canales de Marketing a los que acudir. Desde el SEM⁵ hasta el *email* marketing la oferta es muy variada y sus costes también lo son. A cuáles acudir y en qué momento hacerlo depende en gran medida del presupuesto del que se disponga, aunque también del sector en el que se sitúe el comercio. Sin embargo, el factor común a todos los casos es la necesidad de analizar de forma eficiente el rendimiento y la rentabilidad de cada uno de ellos. Cuando se tienen activas multitud de campañas de diferentes canales esta actividad se hace muy compleja, ocupando (en el caso de ser posible) un gran número de horas a los responsables de marketing y financieros. Como solución a este problema Aptitude desarrolló un avanzado sistema de captura de datos y análisis de los mismos que consta de:

⁵ SEM de sus siglas en inglés Search Engine Marketing, hace referencia al ámbito del marketing online orientado a buscadores.

- Una fase previa a la captura de la información que permite discernir si la visita a la plataforma es humana o no (robot).
- Una segunda fase de captura de la información, donde se identifica por la URL accedida el origen del tráfico y a qué campaña y canal de marketing pertenece. Entonces se procede a registrar la visita en la base de datos junto con toda su información complementaria y a colocar las correspondientes *cookies* en el navegador del cliente. Existen *cookies* de tres tipos:
 - *Cookies* de identificación: permiten identificar inequívocamente al cliente. Para ello es necesario que éste se identifique previamente, bien mediante un inicio de sesión en la plataforma, bien mediante un acceso a la misma a través de un *email* publicitario o *newsletter*.
 - *Cookies* de atribución: Permiten identificar el origen del cliente dentro de la estructura de marketing de la plataforma y atribuir distintos éxitos a los distintos canales en base a un modelo de atribución.
 - *Cookies* de sesión: permiten diferenciar las distintas ocasiones en las que un mismo cliente visita la plataforma y el comportamiento que tiene en cada una de ellas.
- Una tercera fase de análisis de la información obtenida, pudiendo realizarse:
 - Acciones de marketing automáticas, ya que se dispone de toda la información de los gustos e intereses del cliente.
 - Análisis de rentabilidad de las distintas campañas, cruzando la información de los éxitos atribuidos a las mismas con sus costes.
 - Informes de ayuda a la toma de decisiones del departamento de marketing o, incluso, de la dirección de la compañía.

Sistema de envío y análisis de *emails* publicitarios: Si bien, en la parte asociada al análisis de la información, esta herramienta puede incluirse dentro del punto anterior la tratamos como un elemento con entidad propia por diversos motivos. La captación de usuarios y el envío a los mismos de *emails* publicitarios es una de las

palancas⁶ más utilizadas en el mundo del comercio electrónico ya que permite un crecimiento relativamente rápido del volumen de negocio y un control de costes bastante exhaustivo. Cuando se utiliza un modelo de captación de usuarios pagando una cantidad fija por cada uno de ellos (CPL⁷, en función del canal, de la información del usuario recibida, etc.,) es posible analizar, mediante el uso de estudios de cohortes, la rentabilidad de estos usuarios con el paso del tiempo, algo de especial importancia desde el punto de vista financiero. La plataforma Aptshop incorpora además las herramientas necesarias para diseñar, generar y enviar (mediante integraciones con diversas empresas especializadas en el envío de *emails* masivos) las distintas comunicaciones publicitarias, pudiendo aplicar a las mismas todas las segmentaciones necesarias, por ejemplo, enviar sólo a los que hayan abierto la comunicación anterior, o sólo a los que hayan realizado alguna compra en un periodo concreto de tiempo, o que hayan mostrado interés en determinadas áreas, o que hayan realizado búsquedas incluyendo cualquier combinación de criterios.

El resultado obtenido es, como se indicaba anteriormente, una plataforma enfocada principalmente al negocio y a la consecución de objetivos mediante la captura en bruto de información y el análisis posterior de la misma. Sin embargo, como herramienta altamente especializada que es, tiene sus desventajas:

- Sólo es útil en plataformas con alto presupuesto de marketing, puesto que es en estos casos donde este nivel de análisis se hace necesario.
- Su puesta en producción es compleja y requiere de consultoría especializada por parte de Aptitude. Es inviable una descarga y puesta en marcha por parte del usuario como en las plataforma anteriores.
- La utilización correcta de las distintas herramientas requiere de profundos conocimientos, tanto de marketing como del sistema en sí mismo.

⁶ En el argot del marketing, se denomina palanca a las distintas acciones que se pueden realizar para llevar a cabo una estrategia.

⁷ Coste por usuario registrado (en inglés Cost Per Lead)

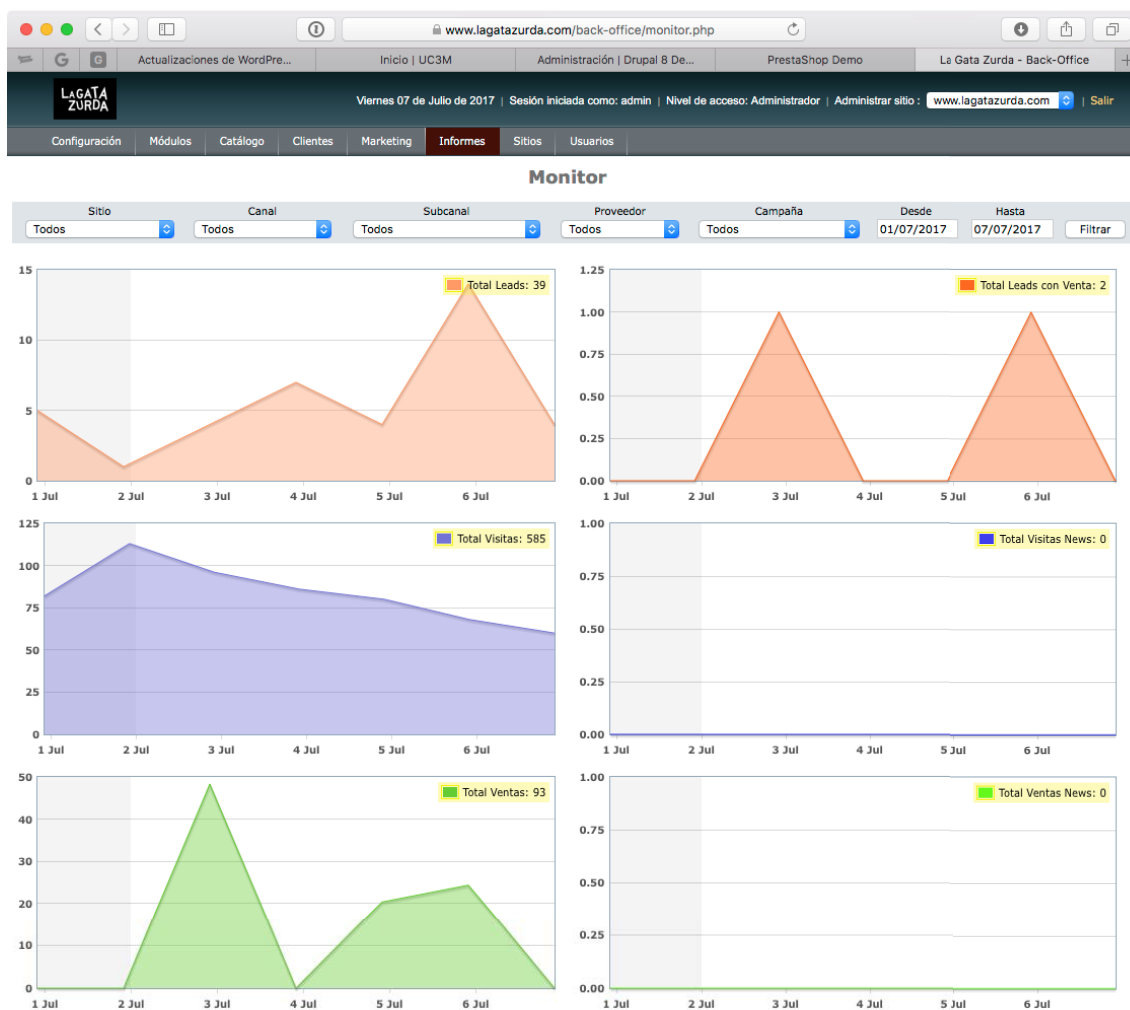


Ilustración 13. Panel de gestión de Aptshop

2.3 Elección de software

Debido a la alta especialización del software, y a la imposible renuncia de su funcionalidad, se decide en junta de socios la continuación con el desarrollo de Aptshop frente a la migración a otras plataformas de código abierto y el desarrollo de un sistema de gestión de contenidos propio a través de un *plugin* en la plataforma. El motivo principal es el no renunciar al sistema de captura de datos y a los sistemas de análisis de los KPIs⁸ específicos de los negocios en los que se encuentra instalado el software, sin olvidarnos de las integraciones con las herramientas de terceros que activan nuestras principales palancas (*newsletters* y notificaciones *push*).

Gracias a estos sistemas, la toma de decisiones relativas a la inversión en canales de marketing activos, se redujo de semanas o meses a unos pocos días. Teniendo en

⁸ Del inglés, Key Performance Indicators son los puntos clave de análisis para conocer el desempeño de un negocio.

cuenta que en nuestros proyectos actuales se pueden invertir hasta 5.000 €/mes en determinados canales, el riesgo de prescindir de estas herramientas ya probadas era muy alto.

3 Análisis del sistema

En este apartado se describen y detallan los casos de uso previstos así como los requisitos, tanto funcionales como no funcionales, que debe cumplir la solución a diseñar. Un mal análisis inicial puede dar lugar a retrasos en la entrega o a un mal resultado de la ejecución de cara al cliente.

3.1 Definición del sistema: Alcance, restricciones y entorno operacional

Como se viene diciendo en los anteriores apartados del documento, el sistema que se pretende construir es un gestor de contenidos para la plataforma Aptshop.

3.1.1 Alcance

Durante el proyecto se desarrollará un *plugin* para la plataforma Aptshop que la proporcione capacidades de sistema gestor de contenidos. Este sistema incluirá la posibilidad de dar de alta nuevas páginas en el sitio web. Vendrá de serie con módulos que permitan modificar la estructura del documento y con otros que permitan añadir contenido. Se incluirán módulos que sustituyan a los módulos por defecto de Aptshop para poder crear desde cero las páginas estándar relacionadas con el comercio electrónico (página de producto, página de categoría, página de resultados de búsqueda, o pagina de carrito de la compra). Se modificará, si fuera necesario, la arquitectura o estructura de Aptshop para dar soporte a este *plugin*. Se prevén cambios en los eventos que lanza la plataforma, para poder insertar funcionalidad en más puntos.

3.1.2 Restricciones

- La aplicación se programará en lenguaje PHP y los módulos deberán crear código HTML5.
- El código JavaScript podrá hacer uso de la librería jQuery que viene incluida en el software Aptshop, pero no de otras externas como MooTools o Zepto.
- Todas las comunicaciones estarán cifradas dado que en algunos casos se manejan datos sensibles y en todos los casos, Google posiciona mejor este tipo de páginas [12].

3.1.3 Entorno operacional

La aplicación Aptshop está preparada para funcionar en sistemas Ubuntu 14.04, por lo que este *plugin* deberá ser compatible con un sistema de las siguientes características:

- Sistema Operativo Ubuntu Linux Server 14.04.5 LTS x64
- Servidor Web Apache 2.4.7
- PHP 5.5.9-1ubuntu4.21
- MySQL Server 5.5.54-0ubuntu0.14.04.1

Las plantillas asociadas a los módulos que se desarrollen deben ser compatibles con las siguientes versiones de navegadores:

Sistema operativo Android: Google Chrome, Firefox y Android Browser v5.0+

Sistema operativo iOS: Chrome, Firefox y Safari.

Sistema operativo Windows 10 Mobile: Microsoft Edge.

Sistema operativo Windows: Chrome, Firefox, Internet Explorer 10+, Microsoft Edge y Opera.

Sistema operativo Mac: Chrome, Firefox, Opera y Safari.

Sistema operativo Linux: Chromium, Chrome y Firefox.

Se utilizará la guía <http://caniuse.com> para determinar si las características de CSS y JavaScript son compatibles con dichos navegadores.

Aunque no debería influir, es importante indicar que Aptshop suele ser instalado en máquinas en la nube de Amazon Web Services y los programadores de Aptitude están habituados a realizar sus pruebas en máquinas VMWare, por lo que el programa desarrollado debe ser compatible con estos sistemas.

3.2 Especificación de requisitos de usuario

En este y en los siguientes puntos se recogen los requisitos a partir de los cuales se diseñará la aplicación. *Los requisitos especifican los criterios que han de ser usados para juzgar la operatividad y los comportamientos específicos de un sistema* [13].

Los requisitos se dividirán en requisitos de usuario y requisitos de software. Hay dos tipos de requisitos de usuario: los requisitos funcionales y los requisitos no funcionales. Para los requisitos funcionales del sistema, se detallarán los casos de uso. Posteriormente se definirán los requisitos de software, los cuales mantendrán una trazabilidad con los requisitos de usuario.

Para la definición de requisitos se utilizará una tabla como la siguiente:

Identificador	Deberá ser único para cada requisito y mantendrá el formato RX-DDD, donde la X se utilizará para identificar el tipo de requisito (RUF: requisito de usuario funcional, RUN: requisito de usuario no funcional) y DDD consistirá en una secuencia de números única para cada tipo de requisito.
Nombre	Nombre del requisito. Deberá ser breve pero descriptivo.
Descripción	Breve descripción del requisito en pocas frases.
Prioridad	Se dividirá en tres posibles valores: alta, media, baja. Tanto las prioridades alta como media deberán ser implementadas. Los requisitos de prioridad baja son los únicos que pueden ser opcionales.
Necesidad	Sus posibles valores son: esencial, deseable y opcional
Fuente	De qué cliente, analista o <i>stakeholder</i> proviene este requisito.
Verificabilidad	Alta o baja en función de lo fácil o complicado que resulta verificar que se está cumpliendo con el requisito.
Estabilidad	Alta: para requisitos de los que no se espera que cambien con el tiempo. Media: para requisitos que es probable que cambien con el tiempo. Baja: para requisitos que, con toda seguridad, se vean modificados en el futuro.

Tabla 1 - Requisito de ejemplo

3.2.1 Requisitos de usuario funcionales

Dado que este proyecto es un proyecto interno de Aptitude Web Services, S.L. los requisitos no han surgido de una petición expresa del cliente, sino de la recolección que los empleados de Aptitude Web Services han venido realizando en base a solicitudes de sus clientes. No obstante, la fuente de los requisitos que se hayan detectado de peticiones de clientes, se establecerá a cliente.

Identificador	RUF-001
Nombre	Acceder al modo edición
Descripción	Un editor que tenga acceso al back-office de la aplicación tendrá la opción de acceder al front-office en modo edición, de modo que aparezcan los controles de gestión del CMS.
Prioridad	Alta
Necesidad	Esencial
Fuente	Analista
Verificabilidad	Alta
Estabilidad	Alta

Tabla 2. Requisito de Usuario RUF-001

Identificador	RUF-002
Nombre	Gestión de páginas de CMS
Descripción	Un editor podrá crear páginas web nuevas siempre y cuando no exista una página previa con la misma URL. De la página se podrán establecer los atributos: título, descripción, palabras clave, URL, nombre, tipo de página, imagen.
Prioridad	Alta
Necesidad	Esencial
Fuente	Cliente
Verificabilidad	Alta
Estabilidad	Alta

Tabla 3. Requisito de Usuario RUF-002

Identificador	RUF-003
Nombre	Estructuras de CMS
Descripción	Los contenidos de las páginas podrán distribuirse en función de unas estructuras predeterminadas, las cuales se podrán combinar para crear superestructuras más complejas. Algunas estructuras serán: columnas, puntos cardinales (ver Ilustración 14. Estructura de puntos cardinales), pestañas y pases de diapositivas.
Prioridad	Alta
Necesidad	Esencial
Fuente	Analista
Verificabilidad	Alta
Estabilidad	Media

Tabla 4. Requisito de Usuario RUF-003

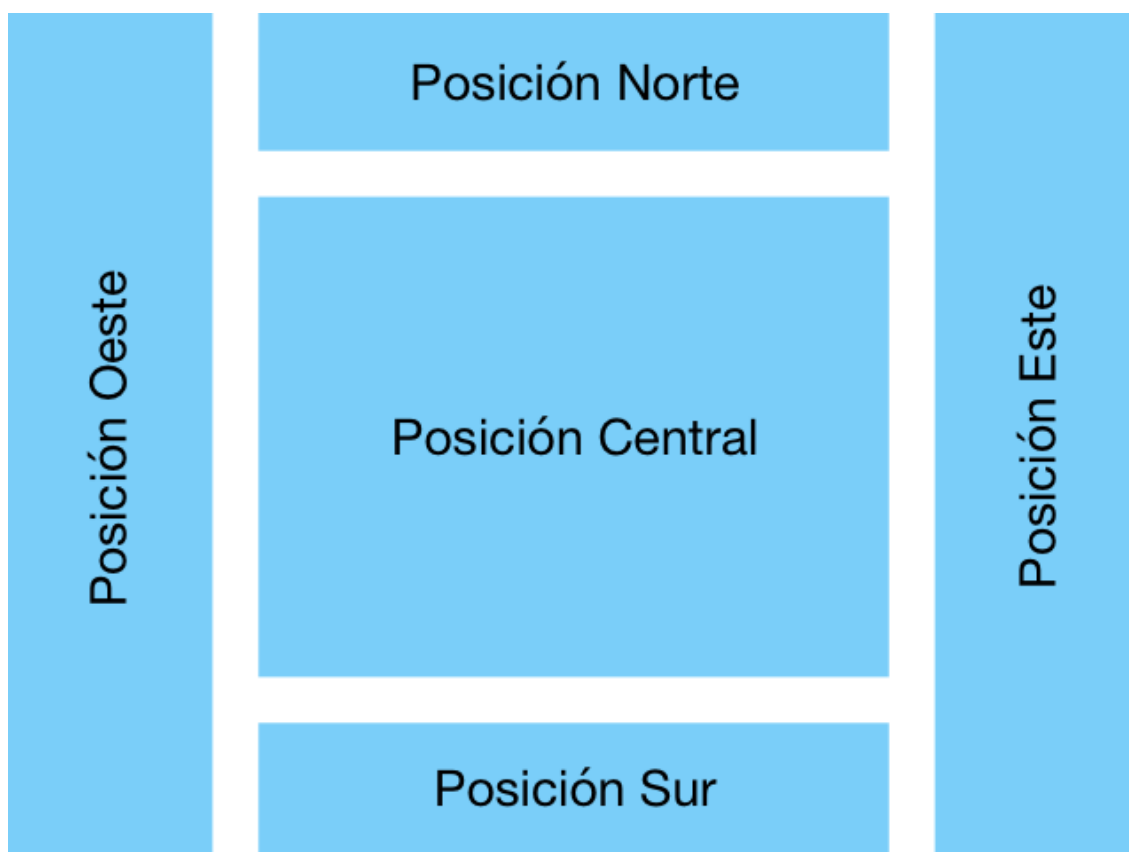


Ilustración 14. Estructura de puntos cardinales

Identificador	RUF-004
Nombre	Gestionar <i>banners</i>
Descripción	Poder dar de alta <i>banners</i> en las distintas secciones de la web, especialmente en la página principal.
Prioridad	Alta
Necesidad	Esencial
Fuente	Cliente
Verificabilidad	Alta
Estabilidad	Media

Tabla 5. Requisito de Usuario RUF-004

Identificador	RUF-005
Nombre	Calendario de módulos de CMS
Descripción	El cliente solicitó que los <i>banners</i> del RUF-004 puedan programarse para que aparezcan o desaparezcan según un calendario. Es más coherente que cualquier módulo de CMS pueda programarse para aparecer y desaparecer en base a un calendario.
Prioridad	Media
Necesidad	Deseable
Fuente	Cliente
Verificabilidad	Alta
Estabilidad	Media

Tabla 6. Requisito de Usuario RUF-005

Identificador	RUF-006
Nombre	Módulos de producto
Descripción	Se podrán incluir en cualquier página, módulo de producto para promocionarlos e incentivar su venta. Por. Ej. Destacar un producto en la página principal del sitio, añadiéndolo a una columna en el lateral de la misma.
Prioridad	Media
Necesidad	Deseable
Fuente	Cliente
Verificabilidad	Alta
Estabilidad	Alta

Tabla 7. Requisito de Usuario RUF-006

Identificador	RUF-007
Nombre	Previsualizar sin publicar
Descripción	El CMS deberá permitir previsualizar cómo quedará la página antes de publicar el contenido.
Prioridad	Alta
Necesidad	Esencial
Fuente	Cliente
Verificabilidad	Alta
Estabilidad	Alta

Tabla 8. Requisito de Usuario RUF-007

Identificador	RUF-008
Nombre	Soporte móvil, tableta y ordenador
Descripción	Se podrán establecer contenidos específicos para cada una de las versiones de la web, independientemente de que las plantillas de los contenidos pudieran ser responsivas.
Prioridad	Alta
Necesidad	Esencial
Fuente	Analista
Verificabilidad	Alta
Estabilidad	Alta

Tabla 9. Requisito de Usuario RUF-008

Identificador	RUF-009
Nombre	Gestionar módulo de testimonios de clientes
Descripción	Existe un módulo en la página web que muestra mediante foto, título y descripción los testimonios de los clientes. Hacerlo editable. No entra en el alcance de este requisito la herramienta de petición de comentarios a clientes.
Prioridad	Baja
Necesidad	Opcional
Fuente	Cliente
Verificabilidad	Alta
Estabilidad	Alta

Tabla 10. Requisito de Usuario RUF-009

Identificador	RUF-010
Nombre	Módulo de imágenes
Descripción	Poder incrustar en la página web imágenes subidas por el editor.
Prioridad	Alta
Necesidad	Media
Fuente	Cliente
Verificabilidad	Alta
Estabilidad	Alta

Tabla 11. Requisito de Usuario RUF-010

Identificador	RUF-011
Nombre	Videos de YouTube
Descripción	Poder incrustar en la página web vídeos de YouTube.
Prioridad	Baja
Necesidad	Opcional
Fuente	Cliente
Verificabilidad	Alta
Estabilidad	Media

Tabla 12. Requisito de Usuario RUF-011

Identificador	RUF-012
Nombre	Gestionar listados de productos
Descripción	Poder incluir un listado de productos que cumplan unas determinadas características relativas a pertenencia a categorías o si tienen asignado un determinado filtro.
Prioridad	Alta
Necesidad	Esencial
Fuente	Cliente
Verificabilidad	Alta
Estabilidad	Baja

Tabla 13. Requisito de Usuario RUF-012

Identificador	RUF-013
Nombre	Gestionar plantillas predefinidas
Descripción	El cliente solicita poder gestionar distintas versiones de la cabecera y el pie en función de la página visitada. Por ejemplo, páginas con cabeceras y pie de página más sencillas en la página de carrito para evitar puntos de fuga del cliente.
Prioridad	Media
Necesidad	Deseable
Fuente	Cliente
Verificabilidad	Alta
Estabilidad	Baja

Tabla 14. Requisito de Usuario RUF-013

Identificador	RUF-014
Nombre	Gestionar formularios de captación
Descripción	Poder incluir en páginas de aterrizaje formularios de captación de clientes.
Prioridad	Alta
Necesidad	Esencial
Fuente	Cliente
Verificabilidad	Alta
Estabilidad	Media

Tabla 15. Requisito de Usuario RUF-014

Identificador	RUF-015
Nombre	Título y descripción para SEO
Descripción	Módulo que suele ir en página superior de la página que incluye el título y la descripción de la misma para mejorar el posicionamiento SEO. Dentro de los diseños de plantillas podrá tener imágenes de fondo.
Prioridad	Baja
Necesidad	Opcional
Fuente	Analista
Verificabilidad	Alta
Estabilidad	Media

Tabla 16. Requisito de Usuario RUF-015

Identificador	RUF-016
Nombre	Decorar módulos. Decoradores genéricos.
Descripción	Se podrán poner títulos, bordes, sombras, botones, enlaces y márgenes a los módulos insertados mediante CMS.
Prioridad	Alta
Necesidad	Esencial
Fuente	Analista
Verificabilidad	Alta
Estabilidad	Baja

Tabla 17. Requisito de Usuario RUF-016

Identificador	RUF-017
Nombre	Mostrar módulos a usuarios.
Descripción	Cuando un usuario visite una página del front-office de Aptshop, se mostrarán en las posiciones correctas todos los módulos que desde el CMS se hayan configurado para esa página y esa fecha.
Prioridad	Alta
Necesidad	Esencial
Fuente	Analista
Verificabilidad	Alta
Estabilidad	Baja

Tabla 18. Requisito de Usuario RUF-016

3.2.2 Requisitos de usuario no funcionales

Identificador	RUNF-001
Nombre	Páginas de CMS en el sitemap.xml
Descripción	Las nuevas páginas que se den de alta a través del CMS deben aparecer en el sitemap.xml si el editor así lo desee.
Prioridad	Alta
Necesidad	Esencial
Fuente	Analista
Verificabilidad	Alta
Estabilidad	Alta

Tabla 19. Requisito de Usuario RUNF-001

Identificador	RUNF-002
Nombre	Acceso sólo para editores registrados
Descripción	Sólo editores registrados en el back-office de Aptshop deben ser capaces de editar el contenido de las páginas.
Prioridad	Alta
Necesidad	Esencial
Fuente	Cliente
Verificabilidad	Alta
Estabilidad	Alta

Tabla 20. Requisito de Usuario RUNF-002

Identificador	RUNF-003
Nombre	Tiempo límite de página
Descripción	El tiempo límite de generación de una página en modo usuario, no en modo edición, no debe ser superior a los 500 ms. En este tiempo no se tiene en cuenta la carga de CSS, JavaScript ni imágenes.
Prioridad	Media
Necesidad	Deseable
Fuente	Cliente Analista
Verificabilidad	Baja
Estabilidad	Baja

Tabla 21. Requisito de Usuario RUNF-003

Identificador	RUNF-004
Nombre	Control de módulos huérfanos
Descripción	Se deben poder controlar los módulos que se han quedado huérfanos. El padre que contenía la posición en el que estaba emplazado puede haber sido borrado. En ese caso, un editor debe ser capaz de detectar dichos módulos huérfanos y reubicarlos.
Prioridad	Baja
Necesidad	Opcional
Fuente	Analista
Verificabilidad	Alta
Estabilidad	Media

Tabla 22. Requisito de Usuario RUNF-004

Identificador	RUNF-005
Nombre	Varios módulos por posición
Descripción	En cada posición de plantilla se podrán añadir 0, 1 o varios módulos. Estos módulos se ordenarán según lo marque el editor. La colocación por defecto será un módulo a continuación de otro, pero esto podría cambiarse en función de la plantilla. (p. ej. en columnas)
Prioridad	Alta
Necesidad	Esencial
Fuente	Analista
Verificabilidad	Alta
Estabilidad	Alta

Tabla 23. Requisito de Usuario RUNF-005

Identificador	RUNF-006
Nombre	Acceso al servidor a través de red
Descripción	Al ser una aplicación web, tanto los usuarios como los editores deben tener acceso a la red en la que se encuentre instalado Aptshop, normalmente a través de Internet.
Prioridad	Alta
Necesidad	Esencial
Fuente	Analista
Verificabilidad	Alta
Estabilidad	Alta

Tabla 24. Requisito de Usuario RUNF-006

Identificador	RUNF-007
Nombre	Resolución mínima editor
Descripción	El editor deberá acceder al CMS con un navegador con una resolución mínima de 1024 px de ancho.
Prioridad	Media
Necesidad	Esencial
Fuente	Analista
Verificabilidad	Baja
Estabilidad	Media

Tabla 25. Requisito de Usuario RUNF-007

Identificador	RUNF-008
Nombre	Agrupación de tipos de módulos de CMS
Descripción	En el editor, a la hora de seleccionar los módulos de CMS de un listado largo, estos aparecerán agrupados para una selección más sencilla. Será conveniente agrupar los módulos que realicen funciones similares. En una primera versión se crearán los grupos estructura y contenido .
Prioridad	Media
Necesidad	Esencial
Fuente	Analista
Verificabilidad	Alta
Estabilidad	Media

Tabla 26. Requisito de Usuario RUNF-008

Identificador	RUNF-009
Nombre	Maquetación compatible con Bootstrap ⁹
Descripción	Aunque Aptshop soporta cualquier tipo de código HTML y CSS, la mayor parte de las plantilla de las que se dispone están realizadas con Bootstrap 3.
Prioridad	Media
Necesidad	Deseable
Fuente	Analista
Verificabilidad	Media
Estabilidad	Baja

Tabla 27. Requisito de Usuario RUNF-009

Identificador	RUNF-010 (Restricción)
Nombre	Lenguajes de programación y librerías
Descripción	La aplicación se programará en lenguaje PHP y los módulos deberán crear código HTML5. En la programación de JavaScript se podrá hacer uso de la librería jQuery pero no de MooTools ni de Zepto.
Prioridad	Alta
Necesidad	Esencial
Fuente	Analista
Verificabilidad	Media
Estabilidad	Alta

Tabla 28. Requisito de Usuario RUNF-010

⁹ Framework para desarrollar proyectos en la web de manera responsiva y con prioridad de móvil frente a ordenadores.

Identificador	RUNF-011 (Seguridad)
Nombre	Conexión https
Descripción	Todas las comunicaciones con el servidor, tanto en el front-office como en el back-office estarán cifradas mediante el protocolo https.
Prioridad	Alta
Necesidad	Esencial
Fuente	Analista
Verificabilidad	Alta
Estabilidad	Alta

Tabla 29. Requisito de Usuario RUNF-011

Identificador	RUNF-012 (Restricción)
Nombre	Entorno operacional
Descripción	La aplicación resultante deberá ser probada y funcionar en la última versión estable de Ubuntu 14.04 LTS. En estos momentos: Ubuntu Server 14.04.5 LTS X64. Servidor Web Apache 2.4.7 PHP 5.5.9-1ubuntu4.21 MySQL Server 5.5.54-0ubuntu0.14.04.1
Prioridad	Alta
Necesidad	Esencial
Fuente	Analista
Verificabilidad	Alta
Estabilidad	Alta

Tabla 30. Requisito de Usuario RUNF-012

Identificador	RUNF-013 (Restricción)
Nombre	Compatibilidad páginas front-office
Descripción	<p>Las plantillas asociadas a los módulos que se desarrollen deben ser compatibles con las siguientes versiones de navegadores:</p> <p>Android: Google Chrome, Firefox y Android Browser v5.0+</p> <p>iOS: Chrome, Firefox y Safari</p> <p>Windows 10 Mobile: Microsoft Edge</p> <p>Windows: Chrome, Firefox, Internet Explorer 10+, Microsoft Edge y Opera.</p> <p>Mac: Chrome, Firefox, Opera y Safari.</p> <p>Linux: Chromium, Chrome y Firefox.</p> <p>Utilizar la guía http://caniuse.com para determinar las características CSS y JavaScript compatibles con dichos navegadores.</p>
Prioridad	Alta
Necesidad	Esencial
Fuente	Analista
Verificabilidad	Baja
Estabilidad	Media

Tabla 31. Requisito de Usuario RUNF-013

Identificador	RUNF-014 (Restricciones)
Nombre	Edición en doble sistema
Descripción	El acceso al modo edición y las acciones <i>CRUD</i> ¹⁰ sobre los módulos y las páginas se llevan a cabo en el sistema de back-office, mientras que los controles que edición que dan acceso a dichas páginas se encuentran en el front-office.
Prioridad	Alta
Necesidad	Esencial
Fuente	Analista
Verificabilidad	Alta
Estabilidad	Alta

Tabla 32. Requisito de Usuario RUNF-014

3.3 Casos de Uso

En este apartado se pasa a describir los casos de uso mediante diagramas y una descripción de los mismos en formato tabular.

Los casos de uso se basan en los requisitos de usuario, y a su vez, durante el análisis de los casos de uso se detectan nuevos requisitos de usuario. De esta forma iterativa se pretenden encontrar todas las funcionalidades que deben quedar descritas en la documentación previa a la fase de diseño.

Es habitual en la modelización de los casos de uso, el agrupar las acciones *CRUD* en un solo caso de uso denominado gestionar. A pesar de que en los requisitos de usuario sí que se ha llevado a cabo esta simplificación, en alguno de los casos de uso siguientes no se habrá hecho debido a la idiosincrasia de los mismos, la cual se explica mejor no agrupando estas acciones.

En el siguiente gráfico se muestran los principales casos de uso que describen las acciones que llevarán a cabo los usuarios en el CMS. Existen dos actores: los administradores/editores de la página y los usuarios de la misma. Los administradores y los editores se corresponden con los perfiles de acceso de la plataforma Aptshop. De cara a los casos de uso los trataremos como uno solo ya que ambos perfiles pueden llevar a cabo las mismas acciones. Para estos usuarios, en adelante los llamaremos

¹⁰ CRUD del inglés *create*, *recovery*, *update* y *delete*, se corresponden con las acciones de añadir, recuperar, actualizar y borrar.

editores, se describen las acciones de gestionar los contenidos de la web, mientras que para el usuario la única actividad detectada es la visualización de la página.

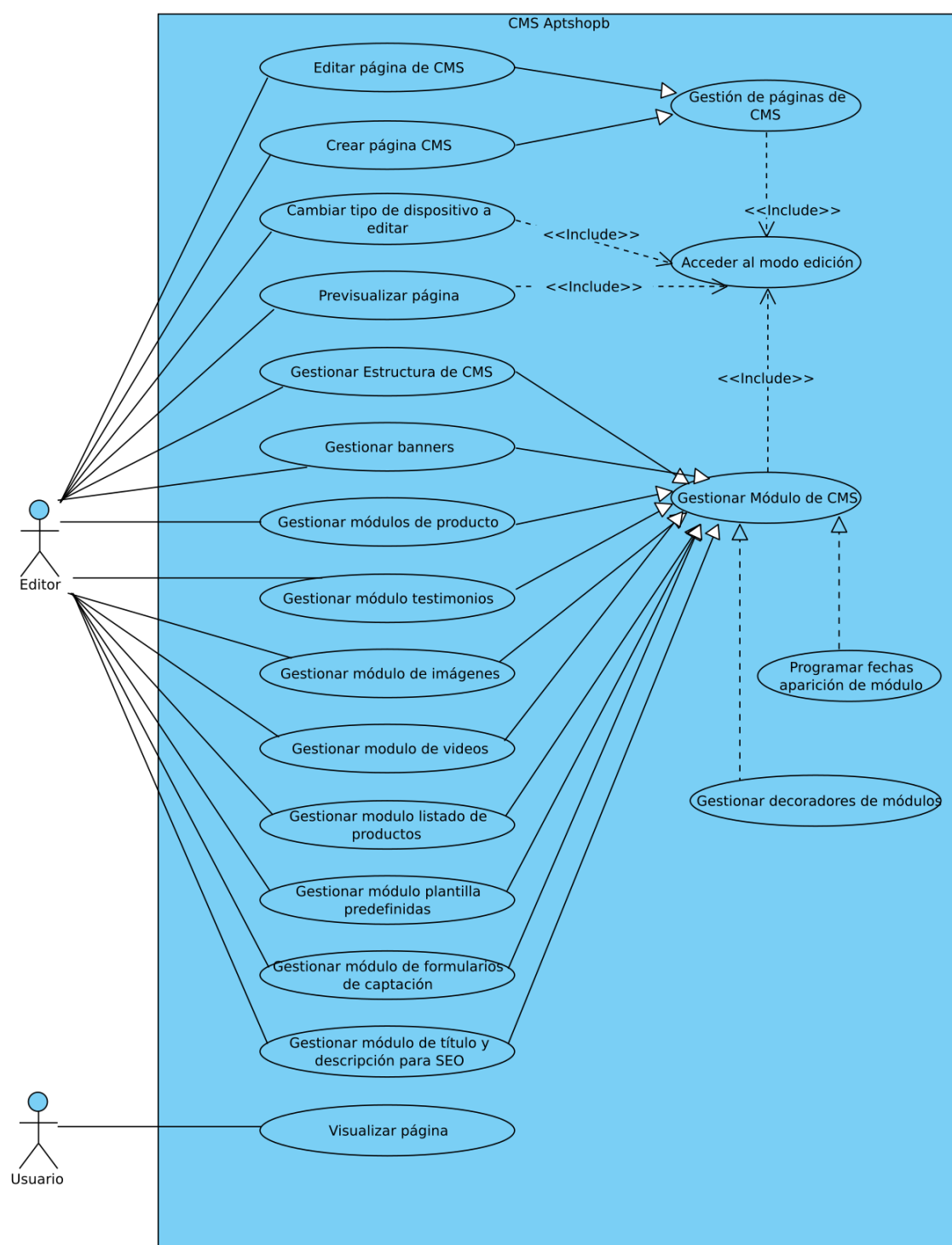


Ilustración 15. Casos de uso

A continuación se detallan cada uno de los casos de uso. No se entrará a describir las acciones que ya están implementadas en la plataforma Aptshop y de las

cuales ya se hace uso. Para describir los casos de uso se utilizarán una tabla como la del ejemplo:

Identificador	Identificador único del caso de uso con el nombre CU-XXX en el que las X son sustituidas por números enteros consecutivos.
Nombre	Un nombre identificativo/descriptivo del caso de uso que permita referirse a él de un modo coloquial.
Objetivo	Describe el propósito del caso de uso.
Precondiciones	Precondiciones que deben cumplirse antes de iniciar el caso de uso.
Postcondiciones	Resultado que se cumplirá después del caso de uso.
Flujo normal	Describe el flujo de acciones o los pasos que se darán para llevar a cabo la ejecución del caso de uso.
Flujo alternativo	Debido a condiciones no habituales, puede existir un flujo alternativo de eventos.
Requisitos de usuario	Enumeración de los requisitos de usuario relacionados con este caso de uso.

Tabla 33. Estructura de tabla para casos de uso

Identificador	CU-001
Nombre	Acceder al modo edición
Objetivo	Permitir al editor entrar en una interfaz de gestión del CMS.
Precondiciones	Editor logado en el back-office de Aptshop.
Postcondiciones	Editor en pantalla de front-office con CMS en modo edición.
Flujo normal	<ol style="list-style-type: none"> 1. Editor accede al back-office de Aptshop 2. Editor pulsa el botón de Gestionar Contenidos
Flujo alternativo	No aplica.
Requisitos de usuario	RUF-001

Tabla 34. Caso de uso CU-001

Identificador	CU-002
Nombre	Crear página CMS
Objetivo	Dar de alta una nueva página en el sistema para permitir dotarla de contenido.
Precondiciones	Editor logado en back-office. CMS en modo edición. Página inexistente.
Postcondiciones	Página nueva creada.
Flujo normal	<ol style="list-style-type: none"> 1. CU-001 2. Editor visita página inexistente. 3. Editor pulsa botón crear página. 4. Editor rellena datos de la página (título, descripción, palabras clave, etc.) 5. Editor pulsa botón guardar. 6. Editor recarga la página.
Flujo alternativo	No aplica.
Requisitos de usuario	RUF-002

Tabla 35. Caso de uso CU-002

Identificador	CU-003
Nombre	Editar página CMS
Objetivo	Modificar las propiedades de una página dada de alta a través del CMS. Las propiedades editables son: título, descripción, palabras clave, estado (activado / desactivado), fecha de activación, fecha de desactivación, URL, nombre para menú e imagen asociada.
Precondiciones	Editor logado en back-office. CMS en modo edición. Visualización de página creada mediante CMS en el front-office.
Postcondiciones	Propiedades de página modificadas
Flujo normal	<ol style="list-style-type: none"> 1. Pulsar botón ‘Modificar página’ 2. Editar propiedades de página. 3. Pulsar botón guardar.
Flujo alternativo	No aplica.
Requisitos de usuario	RUF-002

Tabla 36. Caso de uso CU-003

Identificador	CU-004
Nombre	Cambiar tipo de dispositivo a editar.
Objetivo	Aptshop es un entorno multidispositivo. Distingue entre dispositivo móvil, tableta y ordenador. A través de este caso de uso se permite al editor modificar el dispositivo para el cual se está editando, y se visualizarán los módulos que se hayan asignado a la vista de ese tipo de dispositivo. Por ej. si se cambia a modo móvil no se verán los módulos asignados únicamente a ordenadores.
Precondiciones	Editor logado en back-office. CMS en modo edición. Visualización de página creada mediante CMS en el front-office.
Postcondiciones	Vista en pantalla del front-office de los módulos del dispositivo elegido.
Flujo normal	<ol style="list-style-type: none"> 1. Abrir el desplegable de tipo de dispositivo. 2. Pulsar en el tipo de dispositivo elegido.
Flujo alternativo	No aplica.
Requisitos de usuario	RUF-008

Tabla 37. Caso de uso CU-004

Identificador	CU-005
Nombre	Previsualizar página
Objetivo	Permitir al editor visualizar la página como lo vería un usuario que accediera a ella antes de publicar definitivamente los módulos.
Precondiciones	Editor logado en back-office. CMS en modo edición. Visualización de página creada mediante CMS en el front-office en modo edición.
Postcondiciones	Visualización de página creada mediante CMS en el front-office en modo previsualización, en el cual desaparecen los botones de edición, y se mantiene el botón de salir de previsualización.
Flujo normal	1. Pulsar botón de previsualizar
Flujo alternativo	No aplica.
Requisitos de usuario	RUF-007

Tabla 38. Caso de uso CU-005

Identificador	CU-006
Nombre	Gestionar módulos de CMS
Objetivo	Permite, para los distintos tipos de módulos de CMS, programar fechas de aparición del módulo, gestionar sus decoradores, establecer el estado del mismo (publicado o pruebas), establecer dispositivo o dispositivos en los que será visible (móvil, tableta u ordenador) y la ubicación del mismo en la página (Tipo de página, identificador de página, posición dentro de la página, prioridad dentro de la posición)
Precondiciones	Editor logado en back-office. CMS en modo edición. Visualización de página creada mediante CMS en el front-office en modo edición.
Postcondiciones	Propiedades del módulo modificadas
Flujo normal	<ol style="list-style-type: none"> 1. Pulsar sobre el botón de editar módulo. 2. Modificar las propiedades del módulo. 3. Pulsar botón de guardar.
Flujo alternativo	No aplica.
Requisitos de usuario	RUF-003, RUF-004, RUF-005, RUF-006, RUF-007, RUF-008, RUF-009, RUF-010, RUF-011, RUF-012, RUF-013, RUF-014, RUF-015, RUF-016

Tabla 39. Caso de uso CU-006

Identificador	CU-007
Nombre	Gestionar estructura de CMS
Objetivo	Permitir gestionar un conjunto de estructuras que le provean a la página de huecos o posiciones extra en los que colocar los módulos de manera ordenada. Por ejemplo, en columnas de tamaños iguales, en columnas de tamaños distintos, en pestañas, en diapositivas desplazables, en estructuras de puntos cardinales (Ilustración 14. Estructura de puntos cardinales), etc.
Precondiciones	Editor logado en back-office. CMS en modo edición. Visualización de página creada mediante CMS en el front-office en modo edición.
Postcondiciones	Estructuras creadas, modificadas o borradas según se requiera.
Flujo normal	<ol style="list-style-type: none"> 1. Pulsar en botón de añadir módulo. 2. Seleccionar tipo de módulo: estructura. 3. Seleccionar tipo de estructura entre columnas, filas, pestañas, puntos cardinales o diapositivas) 4. Configurar estructura. 5. Pulsar guardar.
Flujo alternativo	No aplica.
Requisitos de usuario	RUF-003

Tabla 40. Caso de uso CU-007

Identificador	CU-008
Nombre	Gestionar <i>banners</i>
Objetivo	Poder incluir un <i>banner</i> , entendiendo como tal una comunicación comercial en formato imagen con un enlace asociado de destino opcional.
Precondiciones	Editor logado en back-office. CMS en modo edición. Visualización de página creada mediante CMS en el front-office en modo edición.
Postcondiciones	<i>Banner</i> configurado en posición elegida.
Flujo normal	<ol style="list-style-type: none"> 1. Pulsar en botón de añadir módulo. 2. Seleccionar tipo de módulo: contenido. 3. Seleccionar tipo de contenido: <i>banner</i>. 4. Subir imagen. 5. Establecer enlace (opcional) 6. Pulsar guardar.
Flujo alternativo	No aplica.
Requisitos de usuario	RUF-004

Tabla 41. Caso de uso CU-008

Identificador	CU-009
Nombre	Gestionar módulos de producto
Objetivo	Incluir un módulo de producto en una posición de la plantilla o estructura.
Precondiciones	Editor logado en back-office. CMS en modo edición. Visualización de página creada mediante CMS en el front-office en modo edición.
Postcondiciones	Módulo de producto añadido.
Flujo normal	<ol style="list-style-type: none"> 1. Pulsar en botón de añadir módulo. 2. Seleccionar tipo de módulo: contenido. 3. Seleccionar tipo de contenido: producto. 4. Elegir producto. 5. Elegir plantilla. 6. Pulsar guardar.
Flujo alternativo	No aplica.
Requisitos de usuario	RUF-006

Tabla 42. Caso de uso CU-009

Identificador	CU-010
Nombre	Gestionar módulo testimonios
Objetivo	Dar de alta un módulo en el que se reflejen los testimonios de usuarios de la página.
Precondiciones	Editor logado en back-office. CMS en modo edición. Visualización de página creada mediante CMS en el front-office en modo edición.
Postcondiciones	Módulo de testimonios añadido.
Flujo normal	<ol style="list-style-type: none"> 1. Pulsar en botón de añadir módulo. 2. Seleccionar tipo de módulo: contenido. 3. Seleccionar tipo de contenido: testimonios. 4. Configurar un número indeterminado de testimonios, cada uno con fotografía, título y texto. 5. Pulsar guardar.
Flujo alternativo	No aplica.
Requisitos de usuario	RUF-009

Tabla 43. Caso de uso CU-010

Identificador	CU-011
Nombre	Gestionar módulo de imágenes
Objetivo	Incluir una imagen en una de las posiciones de la plantilla o estructura.
Precondiciones	Editor logado en back-office. CMS en modo edición. Visualización de página creada mediante CMS en el front-office en modo edición.
Postcondiciones	Módulo de imagen añadido.
Flujo normal	<ol style="list-style-type: none"> 1. Pulsar en botón de añadir módulo. 2. Seleccionar tipo de módulo: contenido. 3. Seleccionar tipo de contenido: imagen. 4. Subir imagen, establecer texto alternativo y seleccionar plantilla (plantilla de imagen de ancho fijo, plantilla de ancho responsivo, etc.). 5. Pulsar guardar.
Flujo alternativo	No aplica.
Requisitos de usuario	RUF-010

Tabla 44. Caso de uso CU-011

Identificador	CU-012
Nombre	Gestionar módulo de videos de YouTube o Vimeo
Objetivo	Incluir un video de YouTube en un hueco o posición de una plantilla o de una estructura.
Precondiciones	Editor logado en back-office. CMS en modo edición. Visualización de página creada mediante CMS en el front-office en modo edición.
Postcondiciones	Módulo de vídeo añadido
Flujo normal	<ol style="list-style-type: none"> 1. Pulsar en botón de añadir módulo. 2. Seleccionar tipo de módulo: contenido. 3. Seleccionar tipo de contenido: video. 4. Establecer relación de aspecto. 5. Establecer origen del video (YouTube o Vimeo) 6. Establecer identificador del vídeo. 7. Pulsar guardar.
Flujo alternativo	No aplica.
Requisitos de usuario	RUF-011

Tabla 45. Caso de uso CU-012

Identificador	CU-013
Nombre	Gestionar módulo de listado de productos
Objetivo	Incluir un módulo de listado de productos en una posición de una plantilla o de una estructura. El listado de productos se mostrará en función de las categorías o filtros asociados asociados al producto, que hayan sido elegidos por el editor.
Precondiciones	Editor logado en back-office. CMS en modo edición. Visualización de página creada mediante CMS en el front-office en modo edición.
Postcondiciones	Módulo de listado de productos añadido a posición.
Flujo normal	<ol style="list-style-type: none"> 1. Pulsar en botón de añadir módulo. 2. Seleccionar tipo de módulo: contenido. 3. Seleccionar tipo de contenido: listado de productos. 4. Elegir categoría de productos o todas las categorías. 5. Elegir filtro dentro de cada tipo de filtros disponible. 6. Elegir orden de aparición de los productos. 7. Elegir plantilla. 8. Elegir número máximo de productos a mostrar. 9. Pulsar guardar.
Flujo alternativo	No aplica.
Requisitos de usuario	RUF-012

Tabla 46. Caso de uso CU-013

Identificador	CU-014
Nombre	Gestionar módulo de plantillas predefinidas
Objetivo	Poder añadir una plantilla predefinida en un hueco de la página o de otra plantilla. La plantilla puede ser código HTML con o sin huecos.
Precondiciones	Editor logado en back-office. CMS en modo edición. Visualización de página creada mediante CMS en el front-office en modo edición.
Postcondiciones	Plantilla añadida al hueco.
Flujo normal	<ol style="list-style-type: none"> 1. Pulsar en botón de añadir módulo. 2. Seleccionar tipo de módulo: contenido. 3. Seleccionar tipo de contenido: plantilla predefinida. 4. Elegir plantilla del listado. 5. Pulsar guardar.
Flujo alternativo	No aplica.
Requisitos de usuario	RUF-013

Tabla 47. Caso de uso CU-014

Identificador	CU-015
Nombre	Gestionar módulo de formularios de captación
Objetivo	Añadir módulo de formulario de captación en la página.
Precondiciones	Editor logado en back-office. CMS en modo edición. Visualización de página creada mediante CMS en el front-office en modo edición.
Postcondiciones	Formulario de captación añadido en la página.
Flujo normal	<ol style="list-style-type: none"> 1. Pulsar en botón de añadir módulo. 2. Seleccionar tipo de módulo: contenido. 3. Seleccionar tipo de contenido: formulario de captación. 4. Elegir plantilla del formulario. 5. Pulsar guardar.
Flujo alternativo	No aplica.
Requisitos de usuario	RUF-014

Tabla 48. Caso de uso CU-015

Identificador	CU-016
Nombre	Gestionar módulo de título y descripción para SEO
Objetivo	Añadir un módulo que contenga el título y la descripción de la página. Opcionalmente, puede establecerse imágenes de fondo. Este módulo suele incluirse en la parte superior de las páginas para optimizar el SEO.
Precondiciones	Editor logado en back-office. CMS en modo edición. Visualización de página creada mediante CMS en el front-office en modo edición.
Postcondiciones	Módulo añadido con el título y la descripción de la página.
Flujo normal	<ol style="list-style-type: none"> 1. Pulsar en botón de añadir módulo. 2. Seleccionar tipo de módulo: contenido. 3. Seleccionar tipo de contenido: título y descripción para SEO. 4. Elegir plantilla del listado. 5. Pulsar guardar.
Flujo alternativo	<ol style="list-style-type: none"> 1. Pulsar en botón de añadir módulo. 2. Seleccionar tipo de módulo: contenido. 3. Seleccionar tipo de contenido: título y descripción para SEO. 4. Elegir plantilla del listado. 5. Subir imagen de fondo. 6. Pulsar guardar.
Requisitos de usuario	RUF-015

Tabla 49. Caso de uso CU-016

Identificador	CU-017
Nombre	Visualizar página.
Objetivo	Permite a un usuario visualizar una página con todos los contenidos del CMS cargados. Independientemente de que la página haya sido creada mediante el CMS o no, se visualizará su estructura principal con todos los módulos de CMS cargados en sus respectivas posiciones. Se verán los módulos que estén activos para el dispositivo en el que haya entrado el usuario en la fecha en la que haya accedido el usuario.
Precondiciones	Ninguna
Postcondiciones	Página cargada con los módulos visibles.
Flujo normal	1. Usuario visita página web en el navegador.
Flujo alternativo	No aplica.
Requisitos de usuario	RUF-017

Tabla 50. Caso de uso CU-017

Identificador	CU-018
Nombre	Gestionar decoradores de módulos
Objetivo	Permitir, a cualquier tipo de módulo, ser decorado por parte de distintos elementos que modifican el modo en que se ve el módulo. Por ejemplo, un decorador de borde, pondría un borde alrededor del módulo. Un decorador de título, pondría un título antes del módulo y englobaría al módulo en una sección con ese título. Etc.
Precondiciones	Editor logado en back-office. CMS en modo edición. Módulo previamente creado.
Postcondiciones	Módulo decorado con el decorador elegido.
Flujo normal	<ol style="list-style-type: none"> 1. Pulsar botón editar módulo. 2. Seleccionar el decorador o decoradores que se aplicarán al módulo. 3. Configurar los módulos con los parámetros que requieran. 4. Pulsar guardar.
Flujo alternativo	<p>Si el módulo no estuviera creado previamente.</p> <ol style="list-style-type: none"> 1. Pulsar sobre el botón de añadir módulo en una posición. 2. Seleccionar tipo de módulo. 3. Seleccionar subtipo (tipo de contenido, tipo de estructura, etc.) 4. Configurar el módulo. 5. Continúa a partir del paso 2 del flujo normal.
Requisitos de usuario	RUF-016

Tabla 51. Caso de uso CU-018

Para el desarrollo de estos dos últimos puntos **Especificación de requisitos de usuario y Casos de Uso**, se ha iterado en repetidas ocasiones entre ambos puntos [14]. Requisitos de usuario daban a conocer casos de uso y describiendo casos de uso se descubrían nuevos requisitos de usuario. Esta parte del documento suele servir como contrato entre el cliente y la empresa. En este caso, al tratarse de un proyecto interno, no

ha sido así pero sí que ha servido como punto de partida para la redacción de los requisitos de software.

3.4 Especificación de Requisitos de Software

A partir de los requisitos de usuario y los casos de uso, se desarrollara el documento que se entregará al equipo técnico: los Requisitos de Software.

Las principales diferencias entre los requisitos de software y los de usuario son las siguientes:

- Los requisitos de software surgen a partir del análisis de los requisitos de usuario y son más detallados.
- La audiencia de los requisitos de software es mayoritariamente los desarrolladores, frente a la de los requisitos de usuario que es mayormente el cliente.
- En los requisitos de software se pone énfasis en el modelado, métodos formales de organización y es aún más importante su consistencia y completitud.
- En los requisitos de usuario se modela el dominio del problema a resolver mientras que en los requisitos de software se modelan los casos de uso.
- Los casos de uso de los requisitos de usuario hacen hincapié en el nombre, los actores, el objetivo y se describe un escenario básico. Mientras que en los requisitos de software se plantean escenarios alternativos y las pre y post-condiciones.

Los requisitos de software se dividirán de nuevo en requisitos funcionales y no funcionales. Dividiéndose a su vez los no funcionales en consumo de recursos, rendimiento, fiabilidad y disponibilidad, manejo de errores, requisitos de interfaz, restricciones, seguridad.

Dentro de los atributos opcionales que se pueden asociar a los requisitos de software, se opta por incluir riesgo, complejidad, prioridad y necesidad y no se incluyen fuente, estabilidad, coste. Riesgo y complejidad se utilizarán para asignar un perfil adecuado para el desarrollo de la funcionalidad mientras que prioridad y necesidad se utilizarán para priorizarlos en la ventana de tiempo de desarrollo.

3.4.1 Requisitos de software funcionales

Identificador	RSF-001
Nombre	Acceder al modo edición
Descripción	<p>El modo edición es un modo de funcionamiento del front-office en el que se visualiza una página, con todos los módulos cargados (tanto los publicados como los no publicados RSF-007) y con los siguientes botones de edición:</p> <ul style="list-style-type: none"> • Un conjunto de botones de edición, por cada módulo de CMS (editar, eliminar, publicar/despublicar). • Un botón de añadir módulo por cada hueco o posición gestionable de las plantillas. <p>Los editores, son usuarios del sistema de back-office, y la gestión de acciones CRUD de los módulos y páginas de CMS se llevan a cabo en el sistema de back-office. Debe implementarse un modo de acceder al front-office en modo edición desde una sesión del sistema de back-office en la que el editor se encuentre logado.</p>
Prioridad	Alta
Necesidad	Esencial
Riesgo	Alto
Complejidad	Media
Trazabilidad	RUF-001, RUNF-002, RUNF-014

Tabla 52. Requisito de Software RSF-001

Identificador	RSF-002
Nombre	Gestión de páginas de CMS
Descripción	<p>Se creará en el sistema de back-office una formulario de edición de página que permita modificar los parámetros:</p> <ul style="list-style-type: none"> • Título: Metadato de página • Nombre: para menús • Descripción: metadato de página • Palabras clave: metadato de página • Imagen: metadato de página • URL: fragmento de URL • Estado: activada / desactivada • Fechas: desde y hasta • Si se añade al <i>sitemap.xml</i> (RSNF-001) <p>Si se intenta modificar una página que no existe, esta se creará primero y después se editará. Una página dependerá de otra denominada padre formando, una jerarquía. La página raíz será la página principal del sitio. La URL completa de la página se forma con todos los fragmentos de sus ancestros más el suyo propio concatenándolos en orden y separados por el carácter “/”.</p> <p>Desde el front-office en modo edición, aparecerá un botón de crear página cuando nos encontremos en una URL de página inexistente que devuelva un error 404 que nos llevará al editor anterior. De igual modo, aparecerá un botón de editar página en las páginas que hayan sido creadas mediante el CMS.</p> <p>En el front-office modo no edición, se cargará una página con los metadatos que se hayan configurado.</p>
Prioridad	Alta
Necesidad	Esencial
Riesgo	Medio
Complejidad	Baja
Trazabilidad	RUF-002, RUF-001, RUNF-001

Tabla 53. Requisito de Software RSF-002

Identificador	RSF-003
Nombre	Gestión de módulos de CMS
Descripción	<p>Esta parte será común para todos los módulos de CMS, ya sean de estructura o de contenido o de cualquier otro tipo que surja en el futuro.</p> <p>En el back-office se desarrollará un formulario de edición que permita editar 4 puntos:</p> <ol style="list-style-type: none"> 1. Elegir el tipo de estructura, contenido, etc. a insertar. 2. Las opciones de configuración correspondientes con el tipo de módulo elegido. 3. Configurar los decoradores que se deseen asociar al módulo (RSF-017). 4. Una configuración avanzada en la que se incluyan fechas de alta y baja de la publicación (RSF-006, si se visualiza en móvil, tableta y ordenador (RSF-009). <p>En el front-office, en cada hueco, pulsando el botón de añadir módulo, aparecerán los distintos tipos de módulo (contenido, estructura). Al elegir el tipo de módulo, se creará uno del tipo elegido sin configurar y se abrirá el formulario de edición anterior.</p> <p>También en el front-office, sobre un módulo previamente creado aparecerá el botón de modificar módulo que también llevará al formulario de edición anterior.</p>
Prioridad	Alta
Necesidad	Esencial
Riesgo	Alto
Complejidad	Alta
Trazabilidad	RUF-001, RUF-003, RUF-004, RUF-005, RUF-006, RUF-008, RUF-009, RUF-010, RUF-011, RUF-012, RUF-013, RUF-014, RUF-015, RUF-016, RUNF-001, RUNF-005, RUNF-007, RUNF-008

Tabla 54. Requisito de Software RSF-003

Identificador	RSF-004
Nombre	Gestión de estructuras de CMS
Descripción	<p>Aquí se desarrolla el punto 2 del RSF-003:</p> <p>Primero se elegiría el tipo de estructura de entre las siguientes:</p> <ul style="list-style-type: none"> • Columnas • Puntos cardinales (ver Ilustración 14. Estructura de puntos cardinales) • Pestañas • Pases de diapositivas <p>Y posteriormente se configuran una a una. Sólo precisan de configuración las pestañas, para introducir el nombre de las mismas.</p>
Prioridad	Alta
Necesidad	Esencial
Riesgo	Alto
Complejidad	Alta
Trazabilidad	RUF-003, RUF-001, RUNF-005, FUNF-008

Tabla 55. Requisito de Software RSF-004

Identificador	RSF-005
Nombre	Gestión de <i>banners</i>
Descripción	Este requisito amplía el punto 2 del RSF-003: Aptshop dispone de una gestión de <i>banners</i> en el back-office en la cual se crean <i>banners</i> y se asignan a categorías este módulo tendrá dos modos de funcionamiento uno, que se establece cuando el editor selecciona modo de funcionamiento “por defecto”. En el que los <i>banners</i> que se visualizar coinciden con los asignados en el apartado de gestión de <i>banners</i> preexistente. Y otro, en el que el editor puede elegir qué <i>banners</i> se mostrarán. En el caso de elegir más de un <i>banner</i> , se utilizará un carrusel para mostrar un <i>banner</i> después de otro.
Prioridad	Media
Necesidad	Opcional
Riesgo	Medio
Complejidad	Media
Trazabilidad	RUF-004, RUF-001

Tabla 56. Requisito de Software RSF-005

Identificador	RSF-006
Nombre	Calendario de módulos de CMS
Descripción	Este requisito amplía el punto 4 del RSF-003: En la pantalla de edición de módulos del back-office, aparecerán dos fechas (fecha desde y fecha hasta) configurables por el editor, que harán que el módulo activo que se configure, sólo sea visible entre esas fechas. Las fechas se compondrán de año, mes, día, hora, minutos y segundos.
Prioridad	Media
Necesidad	Opcional
Riesgo	Medio
Complejidad	Media
Trazabilidad	RUF-005, RUF-001

Tabla 57. Requisito de Software RSF-006

Identificador	RSF-007
Nombre	Gestión de módulos de producto
Descripción	Este requisito amplía el punto 2 del RSF-003: Cuando se edite un módulo de producto, se permitirá elegir el producto mediante su identificador único y la plantilla con la que se mostrará. Inicialmente se precisan dos plantillas, una para producto pequeño, que se utilizará en los listados de productos y mostrará fotografía, título y precio. Y otra de producto grande, que se utilizará en la página de producto, y tendrá fotografía grande, título, descripción corta, descripción larga, referencia, stock y botón de comprar.
Prioridad	Alta
Necesidad	Esencial
Riesgo	Alto
Complejidad	Media
Trazabilidad	RUF-006, RUF-001

Tabla 58. Requisito de Software RSF-007

Identificador	RSF-008
Nombre	Previsualizar sin publicar
Descripción	Se dispondrá de un botón en el front-office, que al pulsarlo activará un modo edición especial, en el que no se verán los botones de edición (salvo el de salir de modo previsualizar) pero sí que se verán los módulos y páginas no activos (pendientes de publicar). Se verá la página como la vería un usuario normal en el caso de que todos los módulos se encontraran publicados.
Prioridad	Media
Necesidad	Opcional
Riesgo	Bajo
Complejidad	Baja
Trazabilidad	RUF-007, RUF-001

Tabla 59. Requisito de Software RSF-008

Identificador	RSF-009
Nombre	Soporte móvil, tableta y ordenador
Descripción	<p>Este requisito amplía el punto 4 del RSF-003:</p> <p>En la pantalla de edición de módulos del back-office, aparecerá un selector que permita indicar en qué dispositivos aparecerá el módulo que se está configurando. Las opciones serán:</p> <ul style="list-style-type: none"> • Móvil, tableta y ordenador • Móvil • Tableta • Ordenador • Móvil y tableta • Ordenador y tableta <p>En el front-office, en modo edición, aparecerá un botón en el que simular el dispositivo que se está utilizando, permitiendo elegir móvil, tableta u ordenador para visualizar los módulos configurados para cada dispositivo.</p> <p>Un usuario que acceda con alguno de estos dispositivos sólo visualizará los módulos configurados para ser vistos en su dispositivo.</p> <p>Se hará uso de la tecnología de detección de dispositivo ya disponible en Aptshop.</p>
Prioridad	Media
Necesidad	Opcional
Riesgo	Medio
Complejidad	Alta
Trazabilidad	RUF-008, RUF-001

Tabla 60. Requisito de Software RSF-009

Identificador	RSF-010
Nombre	Gestión de módulo de testimonios de clientes
Descripción	<p>Este requisito amplía el punto 2 del RSF-003:</p> <p>Cuando se edite un módulo de “testimonios de clientes”, se permitirá dar de alta cuantos testimonios se deseen. Por cada testimonio se configurará un Título o nombre de cliente, una descripción, relato o testimonio del cliente y una fotografía que acompañe al contenido. Inicialmente se dispondrá de una única plantilla que muestre los testimonios en un pase de diapositivas.</p> <p>En el front-office se visualizará la información dentro de la plantilla elegida.</p>
Prioridad	Baja
Necesidad	Deseable
Riesgo	Bajo
Complejidad	Baja
Trazabilidad	RUF-009, RUF-001

Tabla 61. Requisito de Software RSF-010

Identificador	RSF-011
Nombre	Gestión de módulo de imágenes
Descripción	<p>Este requisito amplía el punto 2 del RSF-003:</p> <p>Cuando se edite un módulo de imágenes se permitirá al editor subir la imagen que desee publicar, un texto alternativo y una plantilla con la que se visualizará. Inicialmente una en la que la imagen se comporte de manera responsiva.</p> <p>En el front-office se visualizará la imagen dentro de la plantilla elegida.</p>
Prioridad	Media
Necesidad	Opcional
Riesgo	Bajo
Complejidad	Baja
Trazabilidad	RUF-010, RUF-001

Tabla 62. Requisito de Software RSF-011

Identificador	RSF-012
Nombre	Gestión de módulo de vídeos de YouTube
Descripción	<p>Este requisito amplía el punto 2 del RSF-003:</p> <p>Cuando se edite un módulo de videos se permitirá al editor elegir la relación de aspecto del vídeo y el identificador de YouTube o Vimeo del vídeo.</p> <p>En el front-office se visualizará vídeo de manera responsiva, manteniendo la relación de aspecto elegida por el editor.</p>
Prioridad	Baja
Necesidad	Opcional
Riesgo	Bajo
Complejidad	Baja
Trazabilidad	RUF-011, RUF-001

Tabla 63. Requisito de Software RSF-012

Identificador	RSF-013
Nombre	Gestión de listados de productos
Descripción	<p>Este requisito amplía el punto 2 del RSF-003:</p> <p>Cuando se edite un módulo, el editor filtrará los productos que se verán en base a las categorías y filtros que se encuentren dados de alta en el sistema Aptshop. Decidirá cuántos productos se verán y en qué orden. Finalmente el editor elegirá la plantilla en la que se visualizarán.</p> <p>En el front-office se visualizará el listado de productos de la tienda que cumplen los criterios especificados, mostrados en la plantilla elegida.</p>
Prioridad	Media
Necesidad	Esencial
Riesgo	Alto
Complejidad	Alta
Trazabilidad	RUF-012, RUF-001

Tabla 64. Requisito de Software RSF-013

Identificador	RSF-014
Nombre	Gestión de plantillas predefinidas.
Descripción	Este requisito amplía el punto 2 del RSF-003: Cuando se edite un módulo, el editor podrá seleccionar una plantilla. En el front-office, se visualizará la plantilla elegida.
Prioridad	Alta
Necesidad	Esencial
Riesgo	Bajo
Complejidad	Baja
Trazabilidad	RUF-013, RUF-001

Tabla 65. Requisito de Software RSF-014

Identificador	RSF-015
Nombre	Gestión de formularios de captación
Descripción	Este requisito amplía el punto 2 del RSF-003: Cuando se edite un módulo, el editor podrá seleccionar una plantilla del formulario. En el front-office, se visualizará la plantilla elegida. La funcionalidad del formulario ya está incorporada en Aptshop, por lo que solo habrá que migrar las plantillas antiguas al nuevo sistema.
Prioridad	Media
Necesidad	Opcional
Riesgo	Medio
Complejidad	Baja
Trazabilidad	RUF-014, RUF-001

Tabla 66. Requisito de Software RSF-015

Identificador	RSF-016
Nombre	Gestión de módulos de título y descripción para SEO
Descripción	<p>Este requisito amplía el punto 2 del RSF-003:</p> <p>Cuando se edite un módulo, el editor podrá seleccionar una plantilla con la que se visualizará el título y la descripción de la página. El formulario de edición permitirá que, opcionalmente, el editor pueda utilizar un título y una descripción distintos a los de la página.</p> <p>En el front-office, se visualizará la plantilla elegida con los datos de título y descripción, etiquetados correctamente (h1 para el título)</p>
Prioridad	Baja
Necesidad	Deseable
Riesgo	Bajo
Complejidad	Baja
Trazabilidad	RUF-015, RUF-001

Tabla 67. Requisito de Software RSF-016

Identificador	RSF-017
Nombre	Gestión de decoradores de módulos.
Descripción	<p>Este requisito amplía el punto 3 del RSF-003:</p> <p>Cuando se edite cualquier módulo, el editor podrá gestionar los decoradores asociados a ese módulo. Estos pueden ser títulos, bordes, sombras, botones, enlaces y márgenes a los módulos insertados mediante CMS.</p> <p>En el front-office, los módulos decorados se verán como tal.</p>
Prioridad	Media
Necesidad	Esencial
Riesgo	Medio
Complejidad	Media
Trazabilidad	RUF-016, RUF-001

Tabla 68. Requisito de Software RSF-017

Identificador	RSF-018
Nombre	Mostrar módulos a usuarios
Descripción	En el front-office, se mostrarán a los usuarios los módulos que el editor haya configurado en la página.
Prioridad	Alta
Necesidad	Esencial
Riesgo	Alto
Complejidad	Media
Trazabilidad	RUF-017, RUF-001

Tabla 69. Requisito de Software RSF-018

3.4.2 Requisitos de software no funcionales

Identificador	RSNF-001
Nombre	Páginas de CMS en el <i>sitemap.xml</i>
Descripción	<p>Este requisito amplía el RSF-002:</p> <p>En la pantalla de edición de página del back-office, una selector que permita elegir la prioridad del <i>sitemap</i> para la página. Eligiendo valores de 0 (no aparece en el <i>sitemap</i>) a 100 (prioridad máxima en el <i>sitemap</i>).</p> <p>En el <i>sitemap</i> se utilizarán las fechas también del RSF-002 y se integrará con el sistema de <i>sitemap</i> ya disponible en Aptshop.</p>
Prioridad	Media
Necesidad	Esencial
Riesgo	Alto
Complejidad	Alta
Trazabilidad	RUNF-001

Tabla 70. Requisito de Software RSNF-001

Identificador	RSNF-002
Nombre	Acceso sólo para editores registrados
Descripción	Como sólo los usuarios de back-office de Aptshop deben ser capaces de editar el contenido de las páginas, se debe acceder al modo edición desde el back-office, generando una clave de acceso única que se almacenará en base de datos y que caducará con el tiempo máximo de sesión. Esa clase se pasará mediante GET, quedando invalidada en el momento en que se de acceso al editor del back-office al modo edición del front-office. Como la edición propia de los módulos se hace en el back-office, se utilizará el sistema de gestión de sesión e inicio de sesión del mismo.
Prioridad	Alta
Necesidad	Esencial
Riesgo	Alto
Complejidad	Baja
Trazabilidad	RUNF-002

Tabla 71 Requisito de Software RSNF-002

Identificador	RSNF-003
Nombre	Tiempo límite de página (500ms)
Descripción	Los módulos como <i>ViewController</i> , como vimos en la arquitectura de Aptshop, se ejecutan en 3 pasos: <i>construct</i> , <i>load</i> y <i>print</i> . Implementar un método de cacheado que evite ejecutar siempre el paso load en los módulos que impidan que se cumpla esta restricción.
Prioridad	Baja
Necesidad	Deseable
Riesgo	Medio
Complejidad	Bajo
Trazabilidad	RUNF-003

Tabla 72 Requisito de Software RSNF-003

Identificador	RSNF-004
Nombre	Control de módulos huérfanos
Descripción	Los módulos huérfanos se cargarán en una posición especial del front-office en modo edición que estará oculta por defecto y que se hará visible a petición del editor.
Prioridad	Baja
Necesidad	Opcional
Riesgo	Bajo
Complejidad	Baja
Trazabilidad	RUNF-004

Tabla 73 Requisito de Software RSNF-004

Identificador	RSNF-005
Nombre	Varios módulos por posición
Descripción	En cada posición de plantilla se podrán añadir 0, 1 o varios módulos. Estos módulos se ordenarán según lo marque el editor. La colocación por defecto será un módulo a continuación de otro, pero esto podría cambiarse en función de la plantilla. (por ej. en columnas)
Prioridad	Alta
Necesidad	Esencial
Riesgo	Bajo
Complejidad	Bajo
Trazabilidad	RUNF-005

Tabla 74 Requisito de Software RSNF-005

Identificador	RSNF-006
Nombre	Acceso al servidor a través de red
Descripción	Al ser una aplicación web, tanto los usuarios como los editores deben tener acceso a la red en la que se encuentre instalado Aptshop, normalmente a través de Internet.
Prioridad	Alta
Necesidad	Esencial
Riesgo	Bajo
Complejidad	Bajo
Trazabilidad	RUNF-006

Tabla 75 Requisito de Software RSNF-006

Identificador	RSNF-007
Nombre	Resolución mínima editor 1024px
Descripción	El editor deberá acceder al CMS con un navegador con una resolución mínima de 1024px de ancho.
Prioridad	Media
Necesidad	Esencial
Riesgo	Medio
Complejidad	Baja
Trazabilidad	RUNF-007

Tabla 76 Requisito de Software RSNF-007

Identificador	RSNF-008
Nombre	Agrupación de tipos de módulos de CMS
Descripción	En el editor, a la hora de seleccionar los módulos de CMS de un listado largo, estos aparecerán agrupados para una selección más sencilla. Será conveniente agrupar los módulos que realicen funciones similares. En una primera versión se crearan los grupos estructura y contenido .
Prioridad	Media
Necesidad	Opcional
Riesgo	Bajo
Complejidad	Media
Trazabilidad	RUNF-008

Tabla 77 Requisito de Software RSNF-008

Identificador	RSNF-009
Nombre	Maquetación compatible con Bootstrap
Descripción	Aunque Aptshop soporta cualquier tipo de código HTML y CSS, la mayor parte de las plantilla de las que se dispone están realizadas con Bootstrap 3.
Prioridad	Media
Necesidad	Deseable
Riesgo	Bajo
Complejidad	Bajo
Trazabilidad	RUNF-009

Tabla 78 Requisito de Software RSNF-009

Identificador	RSNF-010
Nombre	Lenguajes de programación y librerías
Descripción	La aplicación se programará en lenguaje PHP y los módulos deberán crear código HTML5. En la programación de JavaScript se podrá hacer uso de la librería jQuery pero no de MooTools ni de Zepto.
Prioridad	Alta
Necesidad	Esencial
Riesgo	Bajo
Complejidad	Bajo
Trazabilidad	RUNF-010

Tabla 79 Requisito de Software RSNF-010

Identificador	RSNF-011
Nombre	Conexión https
Descripción	Todas las comunicaciones con el servidor, tanto en Font-office como en el back-office estarán cifradas mediante el protocolo https.
Prioridad	Alta
Necesidad	Esencial
Riesgo	Alto
Complejidad	Baja
Trazabilidad	RUNF-011

Tabla 80 Requisito de Software RSNF-011

Identificador	RSNF-012
Nombre	Entorno operacional
Descripción	La aplicación resultante deberá ser probada y funcionar en la última versión estable de Ubuntu 14.04 LTS. En estos momentos: Ubuntu Server 14.04.5 LTS X64 Servidor Web Apache 2.4.7 PHP 5.5.9-1ubuntu4.21 MySQL Server 5.5.54-0ubuntu0.14.04.1
Prioridad	Alta
Necesidad	Esencial
Riesgo	Medio
Complejidad	Baja
Trazabilidad	RUNF-012

Tabla 81 Requisito de Software RSNF-012

Identificador	RSNF-013
Nombre	Compatibilidad páginas front-office
Descripción	<p>Las plantillas asociadas a los módulos que se desarrollen deben ser compatibles con las siguientes versiones de navegadores:</p> <p>Android: Google Chrome, Firefox y Android Browser v5.0+ iOS: Chrome, Firefox y Safari</p> <p>Windows 10 Mobile: Microsoft Edge</p> <p>Windows: Chrome, Firefox, Internet Explorer 10+, Microsoft Edge y Opera.</p> <p>Mac: Chrome, Firefox, Opera y Safari.</p> <p>Linux: Chromium, Chrome y Firefox.</p> <p>Utilizar la guía http://caniuse.com para determinar las características CSS y JavaScript compatibles con dichos navegadores.</p>
Prioridad	Media
Necesidad	Esencial
Riesgo	Alto
Complejidad	Medio
Trazabilidad	RUNF-013

Tabla 82. Requisito de Software RSNF-013

Identificador	RSNF-014
Nombre	Edición en doble sistema
Descripción	<p>El acceso al modo edición y las acciones CRUD sobre los módulos y las paginas se llevan a cabo en el sistema de back-office, mientras que los controles que edición que dan acceso a dichas páginas se encuentran en el front-office.</p>
Prioridad	Alta
Necesidad	Esencial
Riesgo	Alto
Complejidad	Media
Trazabilidad	RUNF-014

Tabla 83. Requisito de Software RSNF-014

3.1 Matriz de trazabilidad de requisitos

	RSF-001	RSF-002	RSF-003	RSF-004	RSF-005	RSF-006	RSF-007	RSF-008	RSF-009	RSF-010	RSF-011	RSF-012	RSF-013	RSF-014	RSF-015	RSF-016	RSF-017	RSF-018	RSNF-001	RSNF-002	RSNF-003	RSNF-004	RSNF-005	RSNF-006	RSNF-007	RSNF-008	RSNF-009	RSNF-010	RSNF-011	RSNF-012	RSNF-013	RSNF-014
RUF-001	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X														
RUF-002		X																														
RUF-003			X	X																												
RUF-004			X		X																											
RUF-005			X			X																										
RUF-006			X				X																									
RUF-007								X																								
RUF-008			X						X																							
RUF-009			X							X																						
RUF-010			X								X																					
RUF-011			X									X																				
RUF-012			X										X																			
RUF-013			X											X																		
RUF-014			X												X																	
RUF-015			X													X																
RUF-016			X														X															
RUF-017																		X														
RUNF-001		X	X																X													
RUNF-002	X																			X												
RUNF-003																					X											
RUNF-004																						X										
RUNF-005			X	X																			X									
RUNF-006																								X								
RUNF-007			X																						X							
RUNF-008			X	X																						X						
RUNF-009																											X					
RUNF-010																												X				
RUNF-011																													X			
RUNF-012																														X		
RUNF-013																															X	
RUNF-014	X																															X

Tabla 84. Matriz de trazabilidad de requisitos

4 Diseño

En este apartado se describirá la arquitectura de Aptshop específica del ámbito en el que va a trabajar el *plugin* de CMS, haciendo hincapié en los puntos de interconexión del *plugin* a desarrollar con la plataforma. Posteriormente se detallará el diseño arquitectónico del *plugin*, tanto en el subsistema de front-office como en el de back-office, así como los modelos relacionales sobre los que se almacenará la información y las librerías basadas en estándares que se utilizarán para resolver el módulo de vista del back-office.

4.1 Arquitectura Aptshop

El programa se compone de dos subsistemas independientes, uno destinado a servir páginas web denominado coloquialmente front-office o simplemente front y otro destinado a administrar el sitio denominado coloquialmente back-office o simplemente back. El CMS de Aptshop deberá integrarse en ambas partes para funcionar. De ese modo, mientras que en el back-office un administrador podrá editar los contenidos, desde el front-office un usuario podrá visualizarlos.

Para una gestión más sencilla de los contenidos se utilizará la estrategia WYSIWYG¹¹ para simplificar la edición. Para ello, aunque la gestión programática se hará sobre el subsistema de back-office, los botones de acceso al gestor se ubicarán en el front-office, permitiendo editar justo lo que se está viendo, y permitiendo al editor hacerse una idea de lo que verán los usuarios cuando se publique. Este punto se verá ampliado en la sección de Diseño.

4.1.1 Arquitectura Front-office

La arquitectura del Front de Aptshop se desarrolló siguiendo la arquitectura de Modelo Vista Controlador (MVC). A continuación se muestra un diagrama simplificado de la arquitectura. Y se explicará, también de manera sencilla el funcionamiento de la misma.

¹¹ WYSIWYG es el acrónimo de What You See Is What You Get. Que en español significa, lo que ves es lo que obtienes, haciendo referencia a que lo un editor ve al modificar los contenidos, será lo que los usuarios vean como resultado final.

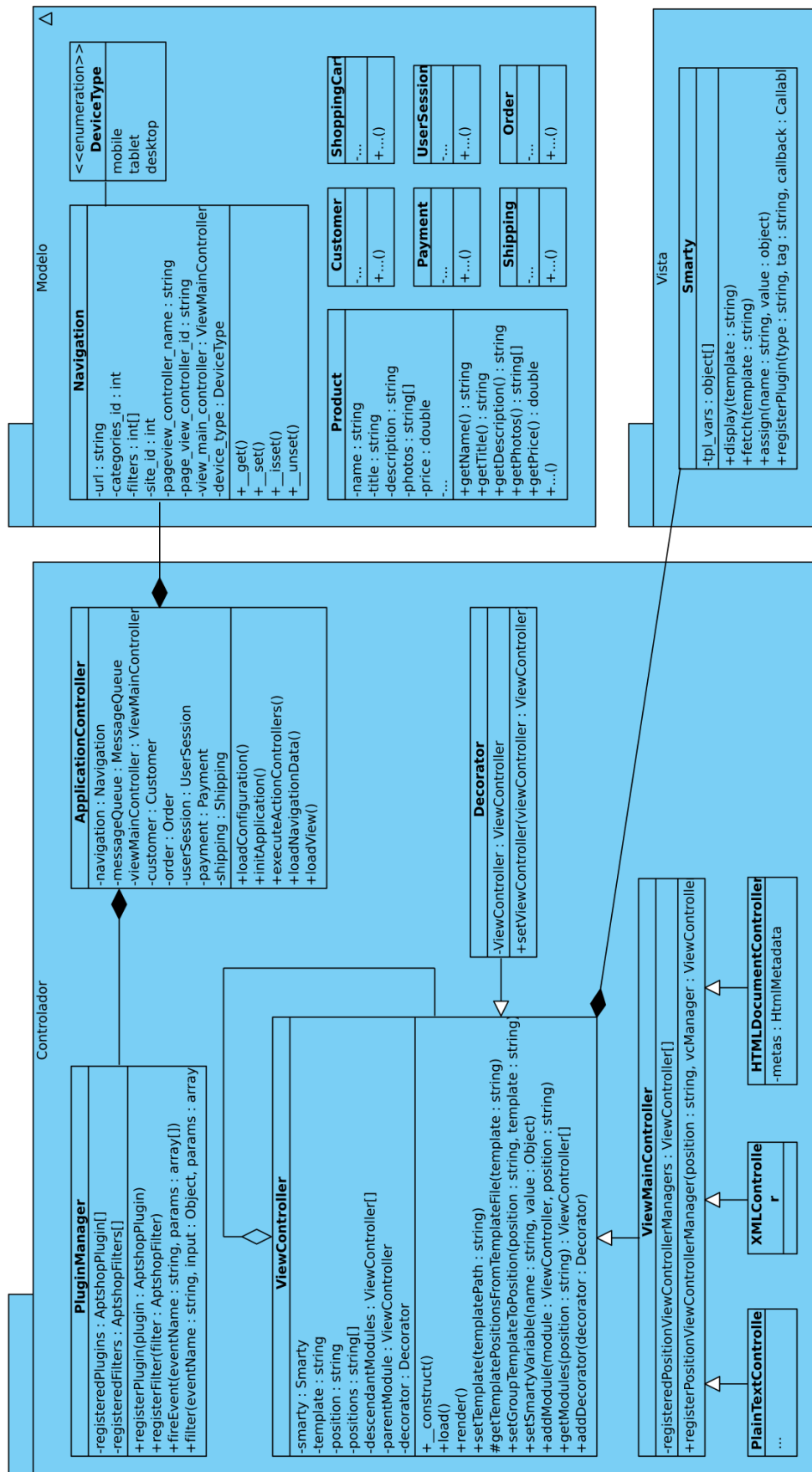


Ilustración 16. Diagrama de clases front-office

En el diagrama podemos ver la arquitectura Modelo Vista Controlador. El programa comienza con la ejecución de una única instancia de ApplicationController, la cual se encarga de inicializar el programa y cargar dependencias (método `initApplication`). Posteriormente, si recibe el comando `action` por parámetros GET ejecuta, mediante el método `executeActionControllers`, las acciones solicitadas. Estos controladores son la principal fuente de modificación del modelo del eCommerce. Se utilizan para que el cliente inicie sesión, la cierre, haga modificaciones en el carrito o modifique sus preferencias. Llegados a este punto y haciendo uso del método `loadNavigationData`, el sistema decodifica la URL y rellena los valores del objeto `Navigation`. En este objeto se almacenan datos relativos a la página a mostrar, los cuales se detallan a continuación:

`url`: Dirección a la que ha accedido el usuario, sin incluir el dominio ni los parámetros GET.

`categories_id`: En caso de encontrarnos en una página de categoría o de filtro, la categoría que se muestra en pantalla.

`filters`: En este atributo se almacena un array con los ids de los filtros que están asociados a la página visitada.

`site_id`: Como ya se ha comentado en el estado del arte, Aptshop es un software multisitio. Se utiliza este atributo para almacenar el id del sitio correspondiente a la visita.

`view_main_controller`: El controlador principal de vista que corresponde a esa URL. Por defecto se utiliza un `HTMLDocumentController`, el cual sirve de marco para cargar contenido HTML en la ejecución. Otros posibles controladores de vista principales soportados son `XMLController` o `PlainTextController`.

`page_view_controller_name`: Indica qué controlador de vista se utilizará para la carga del contenido principal de la página. Controladores habituales son `ModPageProduct` para la página de un producto, `ModMainPage` para la página principal del sitio o `ModFilterPage` para la página de una categoría, un filtro, una búsqueda o una combinación de todos ellos.

`page_view_controller_id`: Es de tipo `string`, y dependiendo del `page_view_controller_name` almacenará un tipo de valor u otro. Por ejemplo, en `ModMainPage` no se utiliza porque es una página única por sitio. En `ModPageProduct` almacena el identificador del producto que se visualizará en pantalla. Y en `ModPaymentProcess` almacena el nombre del paso del proceso de pago.

Finalmente, ApplicationController, con la información del objeto Navigation definida, carga mediante el método loadView el ViewMainController correspondiente. Este a su vez, en caso de estar definido cargará el PageViewController específico.

Todo el subsistema controlador de vista se construye a partir de la clase abstracta ViewController. ViewController es un componente del controlador que se encarga de proveer de los datos del modelo a la vista. Cada ViewController específico contiene un objeto de la librería Smarty, a la cual provee de la plantilla a utilizar, de los datos necesarios para rellenar esa plantilla y de algunas funciones para realizar transformaciones de estos datos. Las plantillas, además de pintar los datos del modelo, pueden contener huecos o posiciones en las que albergar otros ViewControllers. De este modo, la construcción del documento se hace en cascada, partiendo de un documento HTML general, al que se le van incluyendo otros ViewControllers más específicos, que a su vez pueden contener otros ViewControllers internos.

Todos los objetos que hereden de ViewController recorren un mismo camino de ejecución. Se les instancia mediante una llamada al constructor, en el que se debe establecer la plantilla. Posteriormente se llama al método load, que es en el que se obtienen los datos del modelo y ocasionalmente pudieran modificarlo. Si este ViewItem contuviera a otros, llamaría a su método load. Y para finalizar, se llama al método render para obtener el código HTML resultante. Entre la instanciación del objeto ViewItem y la llamada a su método load, el objeto debe haberse incluido dentro de uno de los huecos o posiciones de otro ViewItem. De este modo, cuando el ApplicationController llama al método load o al método render de la clase ViewMainController, al resto de los ViewControllers incluidos se les llama en cascada.

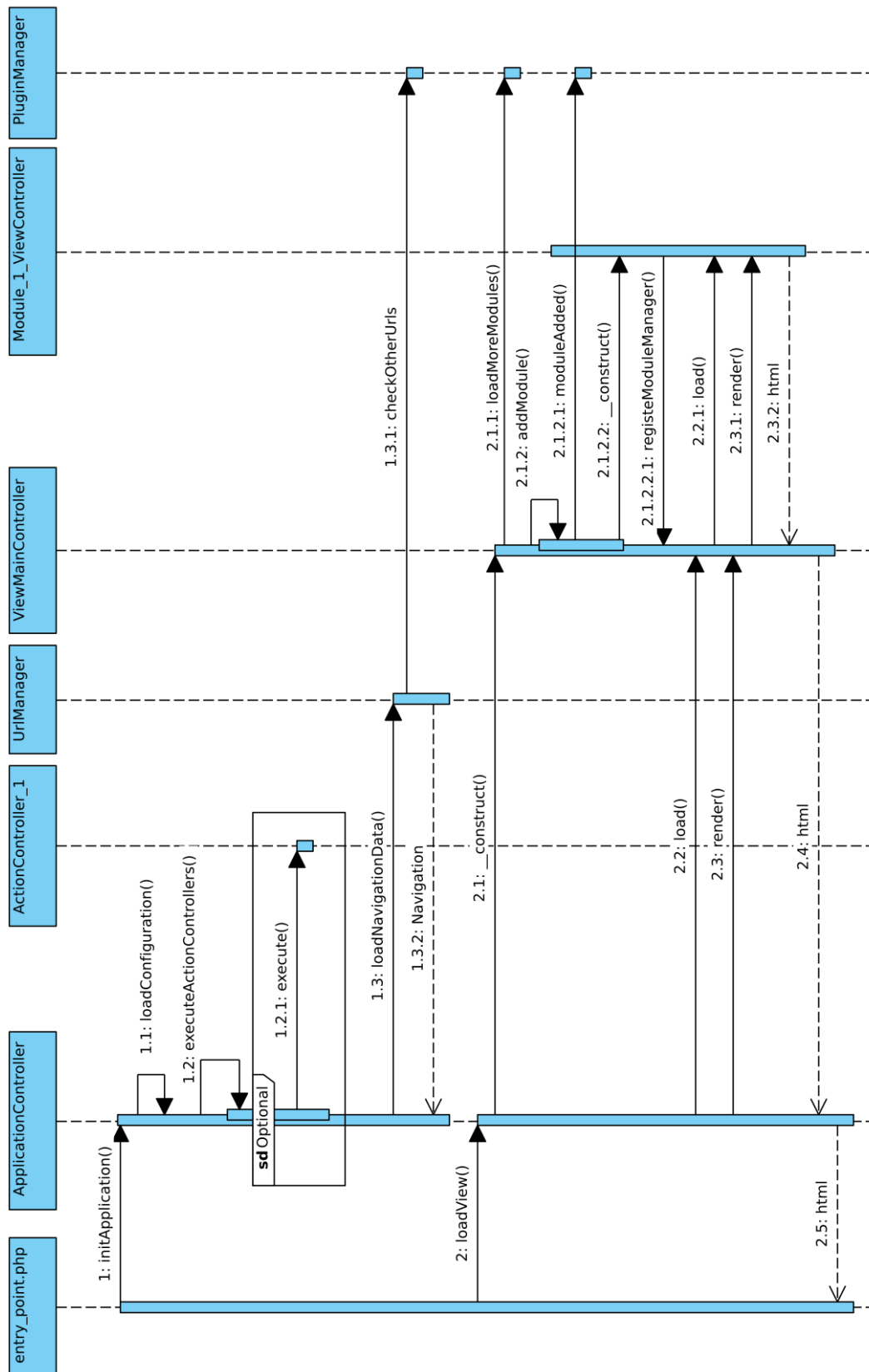


Ilustración 17. Diagrama de secuencia front-office

Existen dos maneras de añadir un ViewController en el hueco o posición de la plantilla de otro ViewController. Conociendo el ViewController de destino, se puede

llamar a su método `addModule`, si existe un hueco o posición con ese nombre en la plantilla, se añadirá el `ViewController`, si no, lanzará excepción. Se pueden incluir varios `ViewController` dentro de cada hueco o posición. Por otro lado, el `ViewMainController` mantiene una relación de los huecos o posiciones de las plantillas con los `ViewController` que los manejan. Entonces, si conocemos el nombre del hueco o posición, pero no el `ViewController` que lo maneja, siempre podemos acceder al mismo a través de `ViewMainController`.

Los `ViewControllers` además implementan el patrón `Decorator`, de manera que cualquier `ViewController` se le puede añadir decoradores que modifican el renderizado final de la plantilla pero que mantienen la interfaz del `ViewController`.

Por otro lado Aptshop dispone de un sencillo pero potente sistema de *plugins*, basado en el patrón `Observer`. Existe un concentrador, denominado `PluginManager` en el cual se registran los *plugins* observadores a la espera de eventos. Una vez se lanza el evento, el `PluginManager` se encarga de notificar al *plugin* para que ejecute su código. Existen dos tipos de eventos, las notificaciones simples, llamadas simplemente eventos, en los cuales se notifica al *plugin* que un evento ha sido lanzado, pasándole un array de parámetros asociado a dicho evento. Por otro lado, las notificaciones de filtro, notifican al *plugin* que se ha lanzado un evento de tipo filtro y se le pasan, además del array de parámetros asociado al evento, el objeto a filtrar por el *plugin*.

Un ejemplo de evento normal sería, que el objeto `ShoppingCart` lance un evento cuando un usuario introduce un producto en la cesta. Un *plugin* de marketing podría escuchar dicho evento con información sobre los productos añadidos al carrito y programar campañas de remarketing de dicho producto en redes sociales.

Un ejemplo de evento de tipo filtro: un *plugin* de compresión de código HTML podría registrarse para escuchar el evento de renderizado de página. Se registraría en dicho evento y recibiría como parámetro el código HTML a imprimir. Tomaría dicho código, lo optimizaría y lo retornaría, de modo que cuando se lo sirva el servidor web ocupe menos espacio.

4.1.2 Arquitectura Back-office

Al contrario que el front-office, el back-office de Aptshop no cuenta con una arquitectura fuertemente definida. Debido a su condición de plataforma heredada, se mantiene igual que en sus orígenes. Se trata de un software hecho mediante programación estructurada. Un controlador se encarga de la puesta en marcha, inicializando los valores de las variables globales principales (sitio, usuario

administrador, etc.). Posteriormente da paso a la carga de un fichero sin estructura definida, la cual queda a elección del programador. No obstante, dispone de acceso a la base de datos mediante un modelo basado en objetos, en los que las clases tienen una equivalencia fuerte con las tablas de la base de datos. Por lo que podríamos decir que utiliza un paradigma mixto de programación estructurada y basada en objetos, pero no orientada a objetos. Se ha integrado el mismo *PluginManager* del front-office dentro del back-office, permitiendo ampliar el funcionamiento del mismo registrando *plugins* o añadidos que escuchan a eventos; no obstante, esta característica no tiene un uso generalizado en el back-office, por lo que hay muchas acciones importantes que no conllevan su correspondiente evento.

4.2 Vista general

Se plantea integrar dentro de la arquitectura descrita en el punto anterior, todo el desarrollo del CMS. Para ello se dividirá el *plugin* en dos. Uno destinado a gestionar todas las labores de front-office y otro para el back-office. El *plugin* del front-office se encargará de mostrar al usuario todos los módulos que se hayan dado de alta desde el back-office, y servirá de punto de acceso mediante botones en el front (cuando esté en modo edición), a las pantallas de back-office de gestión. En el back-office, se gestionará la entrada en modo edición hacia el front-office, y las pantallas de gestión de los módulos y páginas del front-office.

4.3 Front-office

Lo primero a determinar serán los puntos de enlace, relativo a los eventos que el *plugin* escuchará en el front-office. De un modo resumido se puede decir que hay tres acciones principales que el *plugin* realizará en el front-office. Mostrar nuevas páginas dadas de alta a través del CMS, mostrar módulos dados de alta a través del CMS y mostrar los botones de edición cuando se active dicho modo.

Para poder mostrar las páginas de CMS, estas tienen que estar asociadas a una URL. En Aptshop existe un controlador de URLs denominado URLManager (ver Ilustración 17. Diagrama de secuencia front-office). Este controlador lanza el evento *CheckOtherUrls* cuando no es capaz de asociar por si mismo una URL a un *ViewMainController* específico. Haremos que el *plugin* se registre para dicho evento y comprobaremos si la URL pertenece a alguna página creada mediante el CMS.

Para mostrar los módulos que deben cargarse según haya definido el editor, escucharemos al evento *loadMoreModules* que lanza *ViewMainController* al final de su constructor. El *plugin*, al recibir dicho evento, buscará en la base de datos, en base a las

características del objeto Navigation (el cual nos indica las características de la página en la que nos encontramos) y registrará a cada módulo de CMS en el hueco de la plantilla que se le haya asignado en el orden especificado.

Finalmente, para mostrar los botones de edición necesitamos escuchar dos eventos. Utilizaremos también el evento `loadMoreModules` que lanza `ViewMainController` para, por un lado, cargar un módulo que pinte de manera *fixed*¹² en la pantalla, los botones generales de edición (Ver Ilustración 18. Botones de edición CMS en front-office):

- Entrar/salir en modo previsualizar
- Salir del modo edición
- Editar características de la página (título, descripción, etc.)
- Elegir dispositivo de visualización (móvil, tableta u ordenador)

Adicionalmente, se añadirá en cada posición registrada en `ViewMainController` un botón de añadir módulo de CMS.

Por otro lado, se escuchará al evento `moduleAdded` de `ViewMainController`, el cual se lanza cuando se añade un módulo a una posición y tras comprobar que se trata de un módulo de CMS, se le añadirá un decorador que pinta botones que realicen las siguientes acciones sobre dicho módulo (Ver Ilustración 18. Botones de edición CMS en front-office):

- Activar/desactivar
- Editar
- Borrar

¹² *Position fixed* es una característica de CSS que al aplicar sobre una etiqueta HTML, esta se mantiene fija en la pantalla aunque se realice un desplazamiento de la página (*scroll*).

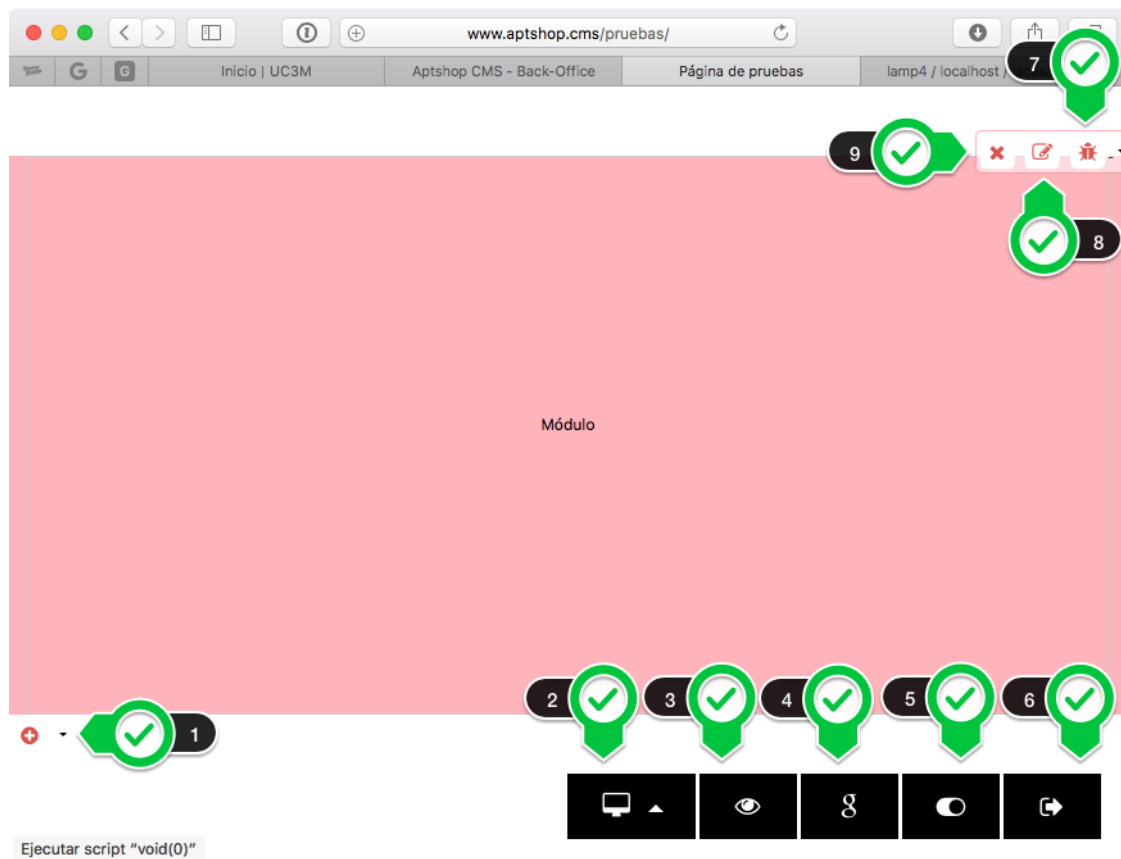


Ilustración 18. Botones de edición CMS en front-office

4.3.1 Modelo

Uno de los requisitos a la hora de implementar este *plugin*, es que el desarrollar módulos simples se pueda hacer rápido y a ser posible sin apenas programación. Para ello se optará por un modelo de datos muy simple, similar a los POJOs de Java. Para ello, el modelo de los datos de los módulos se basará en los tipos más simples de PHP, los cuatro tipos escalares (*boolean*, *integer*, *float* y *string*) y en dos de los tipos compuestos (*array* y *object*). Para la definición de los mismos se hará uso de JSON Schema (compatible con versiones 3 y 4) y quedarán almacenados en un repositorio de módulos (tabla `cms_contents` de la base de datos). La información para instanciar los datos del modelo de los módulos se almacenarán por tanto en JSON dentro de la base de datos, en la tabla `cms_module_instances`.

Las páginas de CMS estarán almacenadas en la tabla `cms_pages` y la asociación de los módulos con las páginas se hará mediante la tabla `cms_modules_to_positions`.

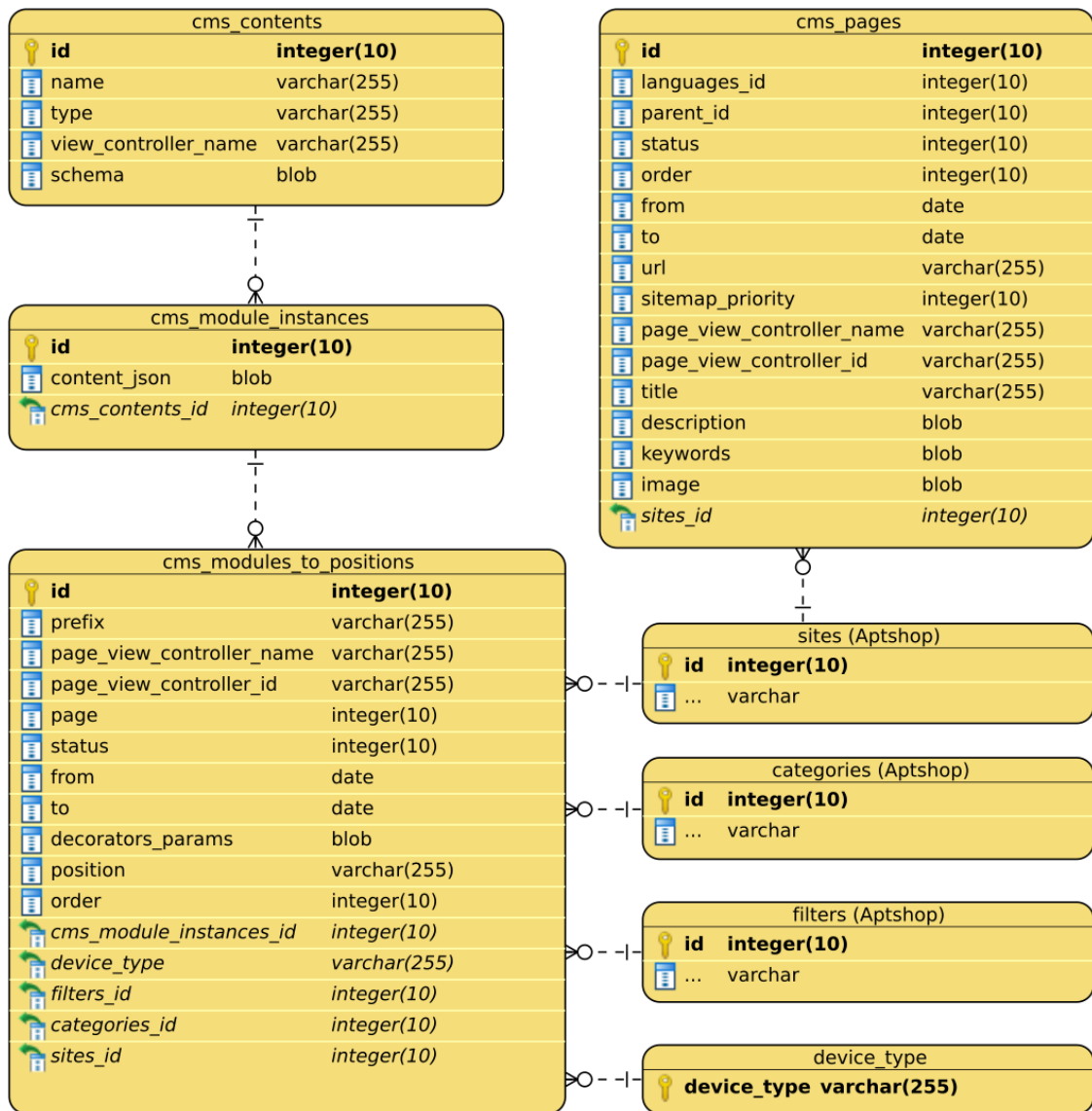


Ilustración 19. Modelo relacional CMS

La entidad *cms_contents* contiene una entrada por cada tipo de módulo de CMS. Estos módulos, su vez se subdividen por el atributo *type* (para satisfacer RSNF-008 Agrupación de tipos de módulos de CMS). De este modo, los distintos módulos de CMS de esta primera versión se clasifican como sigue:

- Módulos de CMS¹³:
 - o *Type* contenido (RSF-003, RSNF-008):
 - Módulo de *Banners* (RSF-005)
 - Módulo de producto (RSF-007)

¹³ La gestión de módulos de CMS está condicionada por los requisitos: RSF-003 Gestión de módulos CMS, RSF-006 Calendario de módulos CMS, RSF-009 Soporte móvil, tableta y ordenador, RSF-017 Decoradores, RSNF-003 Tiempo límite de página.

- Módulo de testimonios de clientes (RSF-010)
- Módulo de imágenes (RSF-011)
- Módulo de vídeo de YouTube (RSF-012)
- Módulo de listado de productos (RSF-013)
- Módulo de plantillas predefinidas (RSF-014)
- Módulo de formulario de captación (RSF-015)
- Módulo de título y descripción para SEO (RSF-003)
- *Type* estructura (RSF-004, RSNF-008)
 - Estructura de columnas
 - Estructura de puntos cardinales
 - Estructura de pestañas
 - Estructura de pase de diapositivas
- *Type* decorador (RSF-017)
 - Decorador de título
 - Decorador de enlace
 - Decorador de bloque (bordes, sombras y márgenes)

Obsérvese que en el modelo relacional, aunque pareciera ser coherente relacionar `cms_pages` con `cms_modules_to_positions`, esto no se ha hecho. Existe una relación, pero no es directa. El contenido de las páginas en Aptshop viene dado por los datos que contiene el objeto `Navigation`. Dos de sus más importantes características son `page_view_controller_name` y `page_view_controller_id`, pero no son las únicas (`filters_id`, `categories_id` y `device_type` son otras de ellas). De este modo, `cms_pages` identifica páginas que se han creado con el CMS y gestionan algunos de estos parámetros. Mientras tanto, existen páginas gestionadas directamente por Aptshop (páginas de categorías, de resultados de búsqueda, de productos) que no están en la tabla `cms_pages`. `cms_modules_to_positions` abarca un espectro mayor de páginas, ya que permite introducir módulos de CMS dentro de todas ellas, y no sólo dentro de las páginas de CMS.

Se adjuntan en el anexo I los modelos JSON Schema que se utilizarán para los distintos módulos.

4.3.2 Controlador

En la Ilustración 16. Diagrama de clases front-office podemos ver la arquitectura del controlador actual simplificada. Este sistema, lo podemos clasificar o dividir en 3 subsistemas. El controlador principal (`ApplicationController`) que dirige la ejecución del

resto de controladores y entre otras, cubre las funcionalidades del software de venta, el controlador de extensiones para ampliar la funcionalidad de Aptshop mediante *plugins* o añadidos (*PluginManager*) y el subsistema de vista, compuesto por las clase *ViewController* y sus subclases. Para integrar este proyecto en Aptshop, se desarrollará un *plugin* denominado *CMSPlugin*. Desde este *plugin* se registrará un *URLManager* específico para interpretar las URLs de páginas de CMS, se cargarán los módulos de CMS almacenados en la base de datos y se decorarán estos módulos con los botones de edición cuando el editor esté modificando la página. Veamos esta integración más detalladamente.

4.3.2.1 *Manejador de URLs de páginas de CMS*

ApplicationController, a su vez, se compone de otros subcontroladores, entre los que destacan *URLManager* y el conjunto de *ActionControllers*. Como ya se ha comentado, *URLManager* toma la URL solicitada por el navegador y genera un objeto *Navigation*, el cual indica en qué sitio nos encontramos, en qué categoría y con qué filtros (si los hubiera), el controlador de vista principal (p. ej. *HTMLDocumentController*) y el controlador de página principal (p. ej. *ModPageProduct* para una página de producto). Además, como ya se comentó en el apartado Front-office, *URLManager* lanza el evento *checkOtherUrls* para permitir a controladores externos comprobar URLs que no ha sido capaz de descifrar por sí mismo.

Se creará la clase *CMSUrlManager* que extenderá de *URLManager* sobrescribiendo el método *decodeURL* en el que se generará un objeto *Navigation* a partir de los datos de la tabla *cms_pages* (ver Ilustración 19. Modelo relacional CMS).

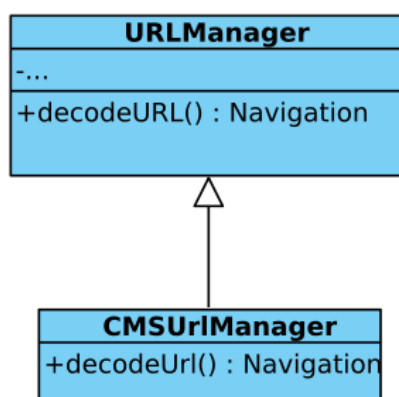


Ilustración 20. Gestión de URLs de CMS

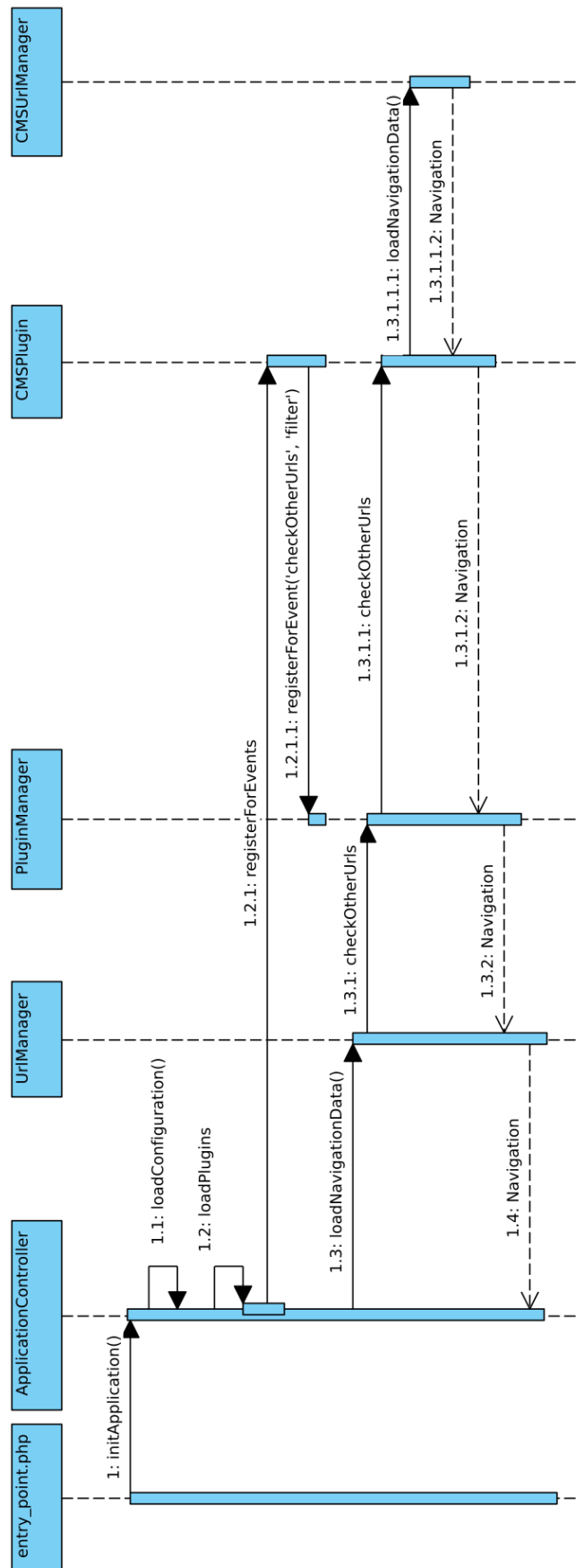


Ilustración 21. Diagrama de secuencia de la Gestión de URLs de CMS

4.3.2.2 Carga de módulos de CMS desde base de datos

El *plugin* *CMSPlugin* dispondrá de un método `loadModulesFromDatabase` que recibirá a un objeto de tipo `Navigation` como parámetro. Este método se encargará de instanciar los distintos módulos de CMS que coincidan con los valores de `Navigation` en cuanto a sitio en el que nos encontramos, tipo de dispositivo, categoría, filtro, `page_view_controller_name` y `page_view_controller_id`. Una vez instanciados, los incrustará en los huecos para los que hayan sido configurados, haciendo llamadas al método `addModule` de `ViewMainController`. `ViewMainController` lleva una relación de todos los módulos cargados en la página con el nombre de los huecos/posiciones que gestionan. En una misma página no puede haber dos posiciones con el mismo nombre.

Los distintos tipos de módulos que se podrán cargar, que están definidos en el apartado *Modelo*, tienen asociado un tipo controlador de vista (`ViewController`) específico. Todos estos nuevos controladores de vista, heredarán de un controlador de vista común `CMSViewController`, que contendrá un constructor común. Este constructor recibirá como parámetro un objeto `stdClass` representativo del JSON con toda la información introducida por el editor (campo `content_json` de la tabla `cms_module_instances`). A parte de llamar al constructor de la clase padre, la cual crea una instancia del componente de vista *Smarty*, comprobará si ese objeto tiene un atributo *template* y se lo asignará a la vista. Además, pasará cada uno de los atributos del objeto a la vista, con los que junto a la plantilla se generará el HTML resultante. Este controlador de vista de CMS, será el más básico de todos, y no debería contener más de 30 líneas de código, pero tendrá la potencia suficiente para dar soporte a módulos que lo único que hagan sea mostrar información que haya dado de alta el editor sin transformar.

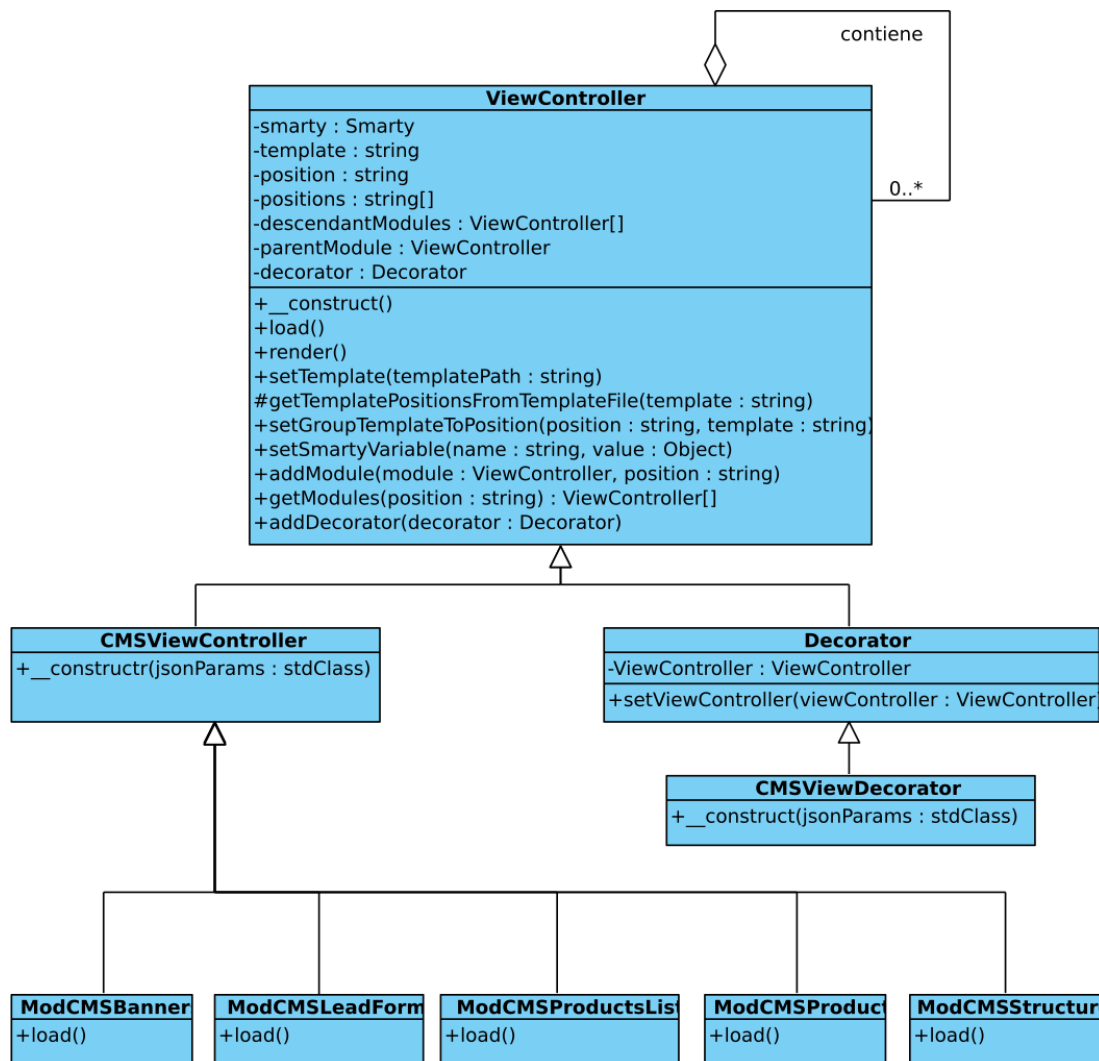


Ilustración 22. Controladores de vistas de CMS

Los distintos módulos serán gestionados a través de los controladores de la imagen según la siguiente tabla:

- Módulos de tipo contenido (RSF-003, RSNF-008):
 - Módulo de *Banners*: ModCMSBanner (RSF-005)
 - Módulo de producto: ModCMSProduct (RSF-007)
 - Módulo de testimonios de clientes: CMSViewController (RSF-010)
 - Módulo de imágenes: CMSViewController (RSF-011)
 - Módulo de vídeo de YouTube: CMSViewController (RSF-012)
 - Módulo de listado de productos: ModCMSProductsList (RSF-013)

- Módulo de plantillas predefinidas: CMSViewController (RSF-014)
- Módulo de formulario de captación: CMSViewController (RSF-015)
- Módulo de título y descripción para SEO: CMSViewController (RSF-003)
- Módulos de tipo estructura (RSF-004, RSNF-008)
 - Estructura de columnas: ModCMSStructure
 - Estructura de puntos cardinales: ModCMSStructure
 - Estructura de pestañas: ModCMSStructure
 - Estructura de pase de diapositivas: ModCMSStructure
- Módulos de tipo decorador (RSF-017)
 - Decorador de título: CMSViewDecorator
 - Decorador de enlace: CMSViewDecorator
 - Decorador de bloque (bordes, sombras y márgenes): CMSViewDecorator

4.3.3 Vista

El apartado de vista será uno de los que menos modificaciones sufra, ya que se delegó la mayor parte de la funcionalidad en la librería externa Smarty, la cual está muy avanzada y se integró a la perfección dentro de Aptshop. No se desarrollará ninguna clase o componente nuevo, pero sí que se realizarán algunas modificaciones. Opcionalmente, el desarrollador puede implementar nuevos añadidos de Smarty.

Como hemos comentado en puntos anteriores, dentro de una página se incluyen ViewControllers, que manejan los huecos o posiciones de sus plantillas en los que se incluyen otros ViewControllers. Estos manejadores de posiciones se registran en el ViewMainController de la aplicación como manejadores de las posiciones que controlan, utilizando el nombre del hueco como clave en la matriz asociativa. Debido a esto, los nombres de los huecos no se pueden repetir dentro de una misma página.

Para solventar esta limitación, a todos los nombres de las posiciones de plantillas controladas por módulos de CMS, se les asignará automáticamente un prefijo compuesto por el identificador de la tabla cms_modules_to_positions seguido de un guion bajo.

4.4 Back-office

Es en el back-office donde se realizarán las operaciones de gestión, aunque no existirá siquiera un modo de navegación que permite acceder a las pantallas, ya que esta tarea se ha encomendado al front-office. Por lo tanto, hay dos temas principales que resolver. La activación del modo edición en el front-office y las pantallas de gestión.

4.4.1 Activación de modo edición

Para el inicio de sesión, se incluirá un enlaces hacia el front-office debidamente modificado para que al pulsar sobre él, el navegador del editor abra una pestaña con el front-office en modo edición, esto es, con los botones de edición del CMS activos. Dicho enlace será de un solo uso y tendrá una caducidad de un minuto. Si se pulsara dos veces en el enlace, solo el primero de ellos daría acceso al modo edición. Así, como si no se utilizara durante un minuto el enlace dejaría de ser válido.

Para ello, se utilizará el objeto AptshopData, el cual almacena de manera persistente objetos PHP en base de datos. Recordemos que según la arquitectura, la base de datos es compartida. El objeto a guardar incluirá una clave autogenerada y una fecha de expiración. Desde el front-office, el plugin de CMS comprobará que se recibe dicha clave por GET se comprobará su validez, invalidará la clave y activará el modo edición en la sesión del front-office.

4.4.2 Pantallas de gestión

Al tratarse de un software heredado, partimos con las limitaciones del mismo, y se debe recordar en este punto que el back-office está desarrollado mediante programación estructurada.

Otro dato a tener en cuenta es que el modelo de datos de los distintos módulos de CMS están estructurados en JSON Schema y que los datos generados se gestionan y almacenan en JSON. Por ello, se establece la herramienta JSON Editor de Jeremy Dorn como *framework* para la implementación de la vista, la cual permite la creación de los formularios a partir de los esquemas de JSON definidos en Anexo I: JSON Schema de los módulos de CMS.

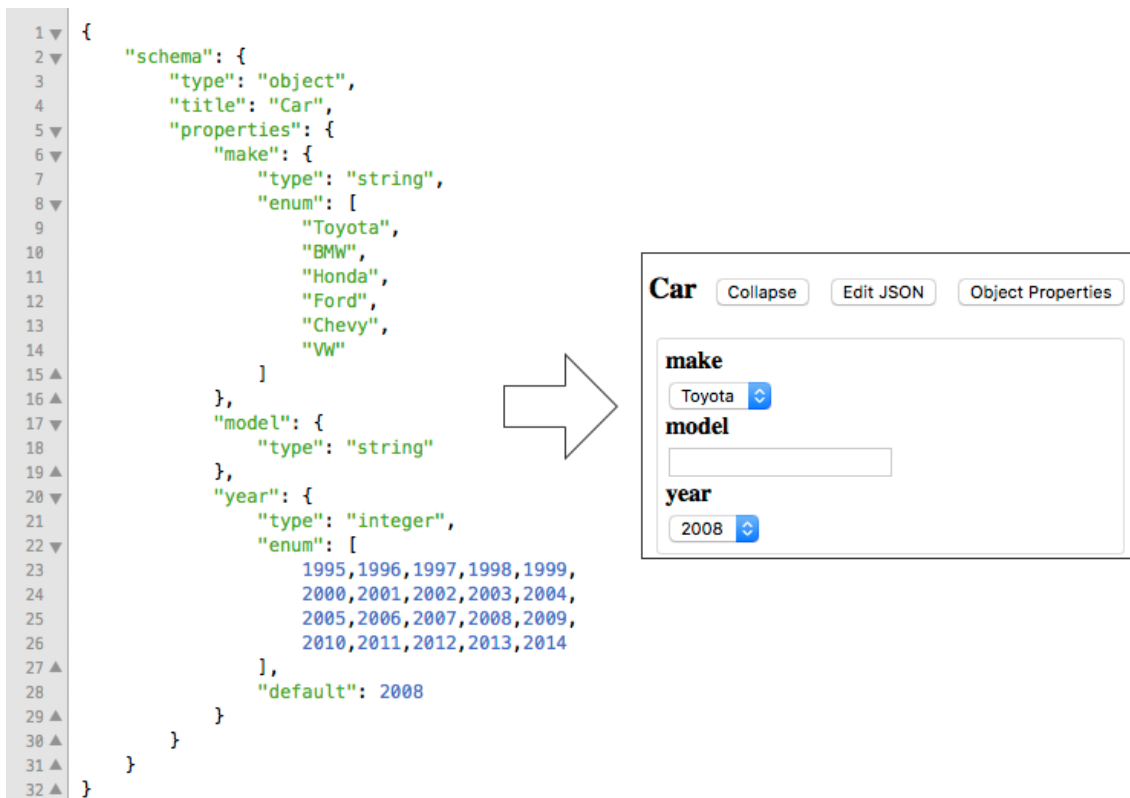


Ilustración 23. JSON Editor por Jeremy Dorn

Los dominios en los que se ejecutan tanto el front-office como el back-office pueden ser distintos, por lo que al back-office se cargará desde el front-office en un *iframe* para evitar errores de *cross site scripting*¹⁴ que se tendría si se utilizara AJAX, sin necesidad de recurrir a técnicas modernas que pueden no funcionar en navegadores antiguos como CORS¹⁵. Tampoco se puede hacer uso de JSONP para estas peticiones, ya que el intercambio de información va a ser pesado y estas llamadas están limitadas al número de caracteres que se pueden pasar por GET al servidor de destino.

Existirán dos pantallas de gestión:

- Pantalla de gestión de módulos de CMS
- Pantalla de gestión de páginas de CMS

Cada una de las pantallas ejecutará un controlador distinto. Estos controladores, primero, realizarán las comprobaciones básicas de cualquier otro controlador de back-

¹⁴ Las peticiones AJAX siguen la política de mismo origen en los navegadores, las cuales son interceptadas por el navegador si el dominio o el protocolo (http/https) es distinto en la página que realiza la solicitud que el destino de la petición AJAX.

¹⁵ Cross Origin Resource Sharing, es un mecanismo de Intercambio de Recursos de Origen Cruzado, el cual permite realizar peticiones AJAX a recursos en servidores de distinto origen de manera segura.

office, las cuales son incluir el script `application_top.php` al principio de la ejecución del controlador y el script `application_bottom.php` al final. Además, estos dos controladores aceptarán peticiones CRUD de los distintos elementos que gestionen. Ambos recibirán como parámetro por GET el objeto Navigation de la página del front-office desde la cual se realizó la petición, codificado en base64. Además, ambas páginas renderizarán la vista haciendo uso del *framework* JSON Editor antes mencionado.

4.4.2.1 Gestión de páginas de CMS

Para instanciar un JSON Editor, se precisa de un esquema JSON y de un valor inicial opcional. El controlador que gestiona las páginas de CMS pasará al editor de JSON un esquema que represente los atributos de la tabla `cms_pages`, en concreto el del Anexo II: JSON Schema de las páginas de CMS.

4.4.2.2 Gestión de módulos de CMS

Para instanciar el editor de JSON para los distintos módulos, se cargarán los esquemas almacenados en la campo `schema` de la tabla `cms_contents`.

Antes de pasar directamente el esquema al editor, se le modificará por unos filtros, los cuales extenderán el esquema dinámicamente. De este modo, el contenido de los esquemas puede modificarse según distintos valores almacenados en base de datos, o según ficheros que existan o no en un directorio. Por ej. se podrá extender el esquema de plantillas predefinidas para que las plantillas se puedan elegir de un listado. De este modo, el esquema inicial:

```
{
  "title": "Plantilla predefinida",
  "type": "object",
  "properties": {
    "template": {
      "options": {
        "templatesDir": "predefinedTemplates"
      },
      "propertyOrder": 2
    }
  }
}
```

Pasaría a convertirse de manera dinámica en:

```
{
  "title": "Plantilla predefinida",
  "type": "object",
  "properties": {
    "template": {
      "options": {
        "templatesDir": "predefinedTemplates",
        "enum_titles": [
          "AhorraUn70YFinanciacionGratis.tpl",
          "BannerTienesUnPresupuesto.tpl",
          "ModPageDefaultHeader.tpl",
          "ModPageLandingHeader.tpl"
        ]
      }
    }
  }
}
```

```

    },
    "propertyOrder": 2,
    "title": "Plantilla",
    "type": "string",
    "enum": [
      "predefinedTemplates/AhorraUn70YFinanciacionGratis.tpl",
      "predefinedTemplates/BannerTienesUnPresupuesto.tpl",
      "predefinedTemplates/ModPageDefaultHeader.tpl",
      "predefinedTemplates/ModPageLandingHeader.tpl"
    ]
  }
}
}

```

Los extensores de esquema que se implementarán son:

- Extensor de categorías: Carga dinámicamente un conjunto de **categorías** de la base de datos y lo incrusta en el JSON schema.
- Extensor de productos: Carga dinámicamente un conjunto de **productos** de la base de datos y lo incrusta en el JSON schema.
- Extensor de filtros: Carga dinámicamente un conjunto de **filtros** de la base de datos y lo incrusta en el JSON schema.
- Extensor de plantillas: Carga dinámicamente un conjunto de **plantillas** desde una ruta de directorio y lo incrusta en el JSON Schema.
- Extensor de posiciones dinámicas: Crea una estructura de JSON schema compleja de manera programática a partir de unos parámetros.

5 Construcción de la solución

En este apartado se muestran las pantallas resultantes de la implementación de la solución. Al principio se verán las pantallas principales de acceso al editor y de creación y edición de páginas de CMS. Posteriormente se mostrarán un conjunto de pares: pantalla de edición y pantalla de resultado.

5.1 Acceso al modo edición

Una vez se accede al back-office, aparece un nuevo elemento en la página principal que permite iniciar sesión en modo edición en el front-office.

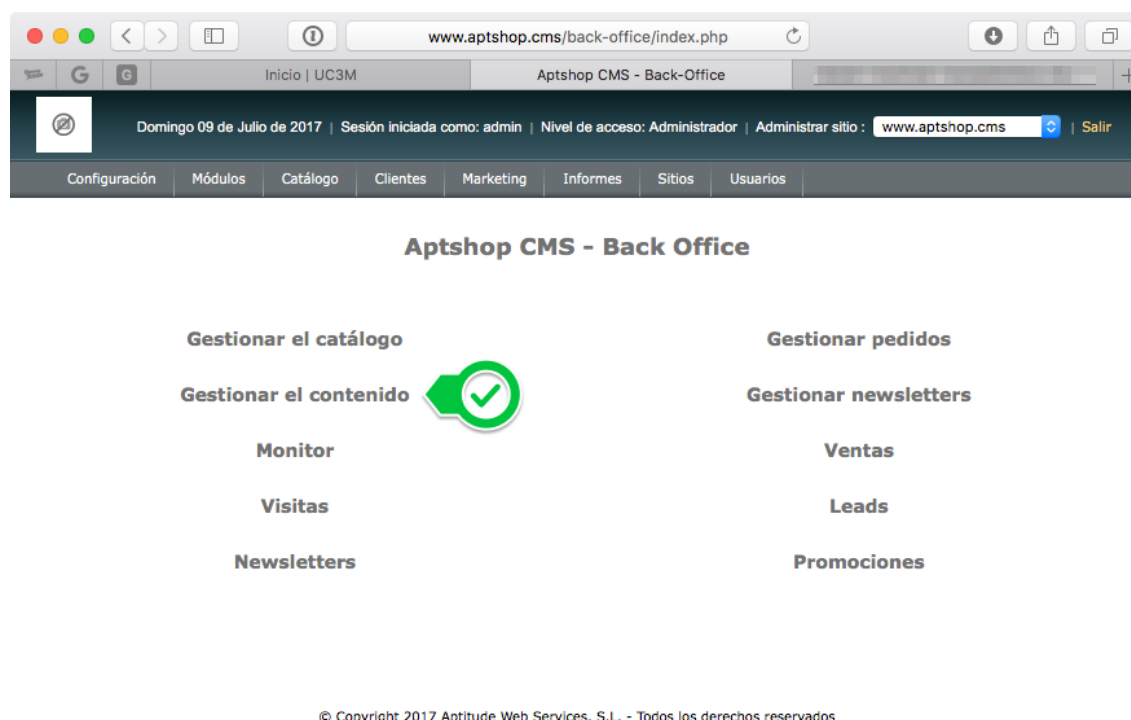


Ilustración 24. Acceso al modo edición desde el back-office

Al cargar la página, se creará un enlace único para el botón “Gestionar el contenido” que enlace con el front-office y de acceso al modo edición.

5.2 Front-office en modo edición

Una vez se acceda al front-office en modo edición, se verá en la parte inferior derecha un conjunto de botones generales del editor (2, 3, 4, 5, 6). También habrá un botón de añadir módulo por cada hueco o posición de las plantillas, y un botón de edición de módulo en cada módulo de CMS (7, 8, 9).

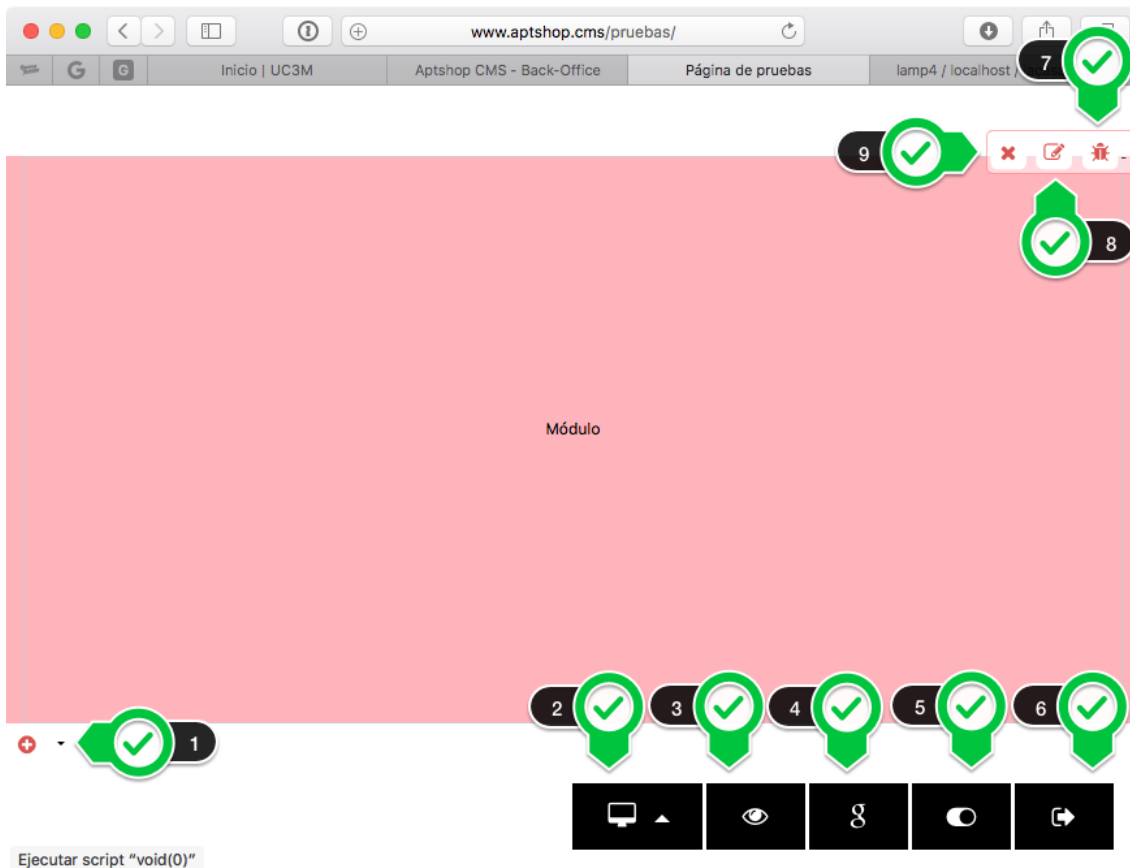


Ilustración 25. Front-office en modo edición

1. Un símbolo más en cada hueco de las plantillas. Permite añadir nuevos módulos en los huecos. Al pulsar se abrirá un desplegable que nos permita elegir el tipo de módulo a insertar. Ver Ilustración 26. Insertar nuevo módulo.
2. Permite elegir el modo de vista entre desktop, tablet o mobile, para ver los módulos insertados para cada uno de los distintos tipos de dispositivos.
3. Por defecto, los botones 7, 8 y 9 no se ven hasta que se pase el ratón por encima del módulo a editar o se pulse este botón nº 3. Esto es así porque se prevé que exista un gran número de botones de edición, y queremos evitar ensuciar demasiado la pantalla.
4. El botón con el símbolo de google nos permite editar los parámetros de página destinados al SEO (título, descripción, etc.). Abrirá una ventana de edición de back-office de edición de página de CMS. Ver Ilustración 27. Pantalla de edición de página.
5. Permite previsualizar la página tal y como la vería un usuario, sin ningún botón del CMS salvo el de volver al modo edición.

6. Salir del modo edición.
7. Por defecto los módulos se activan en modo pruebas, esto es, que el editor los ve, pero los usuarios no. El icono de este botón nos indica el estado actual y nos permite, mediante desplegable, activarlo, dejarlo en modo pruebas o desactivarlo totalmente.
8. Botón de acceso a la pantalla de modificación del módulo.
9. Botón de borrar módulo.

El pulsar sobre el botón de añadir módulo, se desplegará un conjunto de opciones que permiten al editor seleccionar el tipo de módulo que desee añadir.

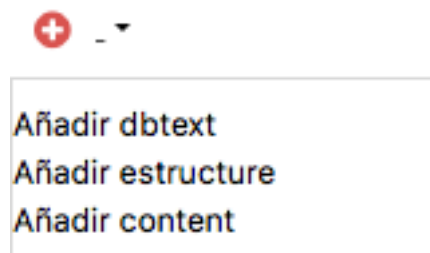


Ilustración 26. Insertar nuevo módulo.

5.3 Back-office, gestión de página de CMS

Tanto en las páginas de CMS como en las páginas que den error 404 (página inexistente) aparecerá el botón 4 de la Ilustración 25. Front-office en modo edición. En las primeras, permitirá modificar los metadatos de la misma y en las segundas permitirá crear una página nueva de CMS y modificar los metadatos.

cms_page

cms_pages_title

descripcion code

```

1 Descripción de la página de pruebas. Aquí caben
2 varias líneas.
3

```

cms_pages_status **cms_pages_order** **cms_pages_sitemap_priority**

Fecha desde (AAAA-MM-DD HH:MM:SS) **Fecha hasta (AAAA-MM-DD HH:MM:SS)**

cms_pages_url **cms_pages_menu_name**

cms_pages_template_master_name **cms_pages_template_master_id**

cms_pages_keywords **cms_pages_image**

[ver imagen](#)

Ilustración 27. Pantalla de edición de página

Título: Se utiliza como metadato título de la página HTML.

Descripción: Metadato descripción de la página HTML.

Keywords: Metadato de palabras clave de la página HTML.

Image: Metadato de imagen en la página HTML, se tiene en cuenta a la hora de compartir la página en redes sociales.

Estado: Activado (los usuarios pueden acceder a la página) y desactivado (los usuarios no pueden acceder a la página. Error 404).

Order: En caso de aparecer la página en un listado (p. ej. menú), se ordenarán según este atributo.

Sitemap priority: Si es 0, no aparecerá en el sitemap.xml del sitio. Cualquier otro valor, de 1 a 100 marcará la prioridad en dicho fichero.

Fecha desde, fecha hasta: Define el intervalo de fechas entre las que se podrá acceder a la página.

5.4 Módulo de *banners*

El módulo de banners muestra un conjunto de imágenes con un enlace asociado cada una. Permiten, de una manera visual, dirigir el tráfico a otras páginas del sitio (o de otros sitios). Se usa a menudo en las páginas principales de las tiendas para destacar productos de alto interés en ese momento. El modo elegido para incorporar más de un banner ha sido utilizando un pase de diapositivas o carrusel.

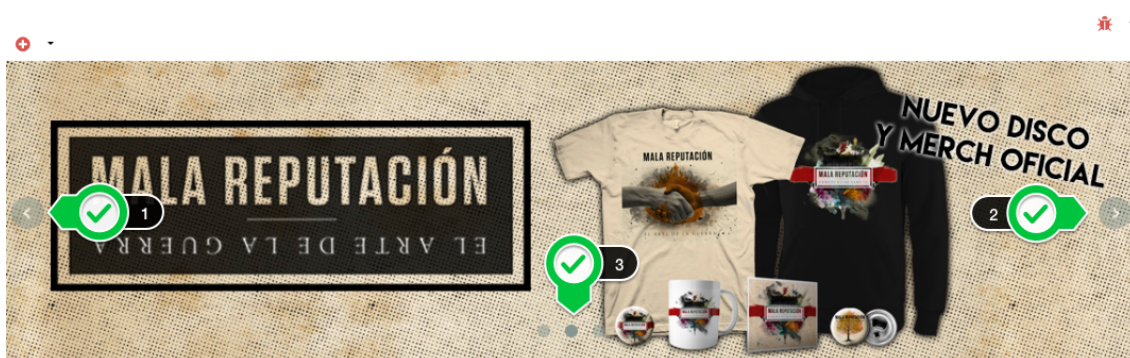


Ilustración 28. Módulo de banners (front-office)

1. Ver *banner* anterior.
2. Ver *banner* siguiente.
3. Paginación.

En la siguiente pantalla podemos ver el módulo de edición de banners del back-office. Un usuario se limitará a elegir el tamaño de banner que desee incorporar, y seleccionar los banners dentro de un listado que se corresponde con las campañas activas de Aptshop.

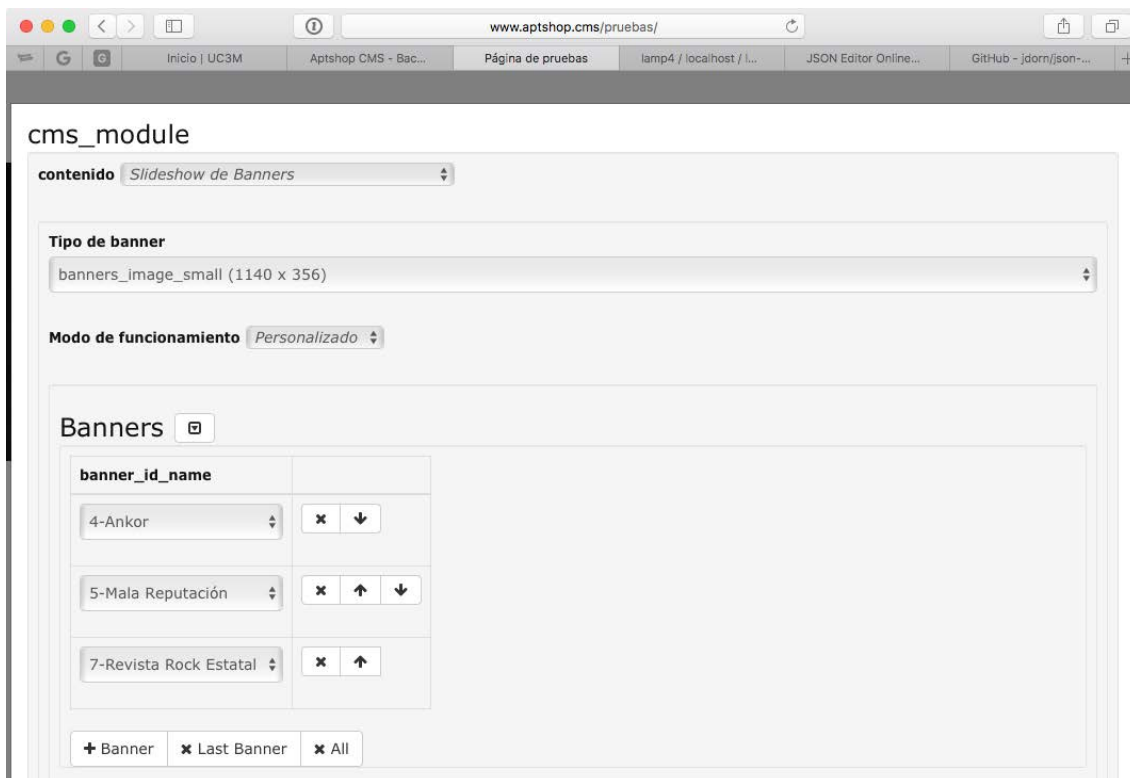


Ilustración 29. Módulo de banners (back-office)

5.5 Módulo de producto

El módulo de producto es el más importante del CMS. Toma como entrada un identificador de producto y muestra en pantalla una representación visual del mismo, definido por la plantilla elegida. Se han desarrollado varias plantillas para este módulo. ModOfferSmall, es la que vemos en el ejemplo y se trata de una representación de un producto apta para listados de productos, o inserción en huecos pequeños. Existe un ModOfferBig para insertar en la página de producto, que incluye selector de variantes (por ej. talla) y botón de comprar, y otras plantillas con fines distintos. Crear una nueva plantilla de producto resulta muy sencillo para un diseñador o programador que esté familiarizado con HTML y CSS.



Blink 182 - 2Vínilo Blink 182 (Lp)
33,99€

Ilustración 30. Módulo de producto (front-office)

Este módulo dispone de numerosas plantillas para representar el producto de diversas maneras.

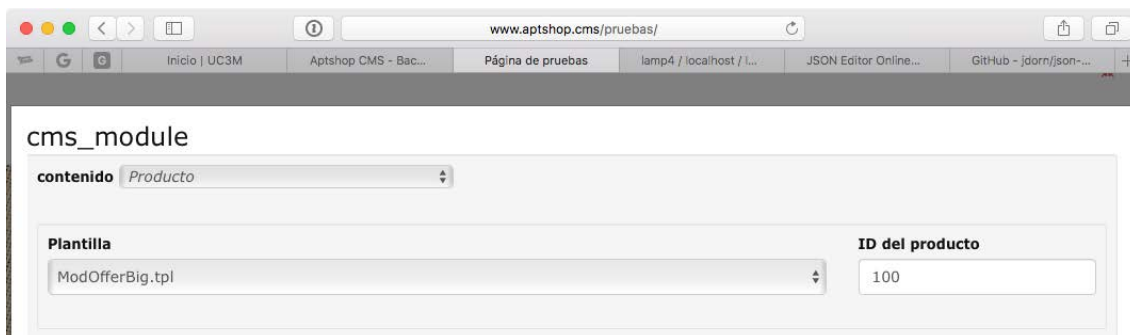


Ilustración 31. Módulo de producto (back-office)

5.6 Módulo de testimonios de clientes

Para transmitir cercanía y confianza a los usuarios, nos valemos de los testimonios de clientes que han acabado satisfechos con nuestros servicios y los promocionamos en la página.

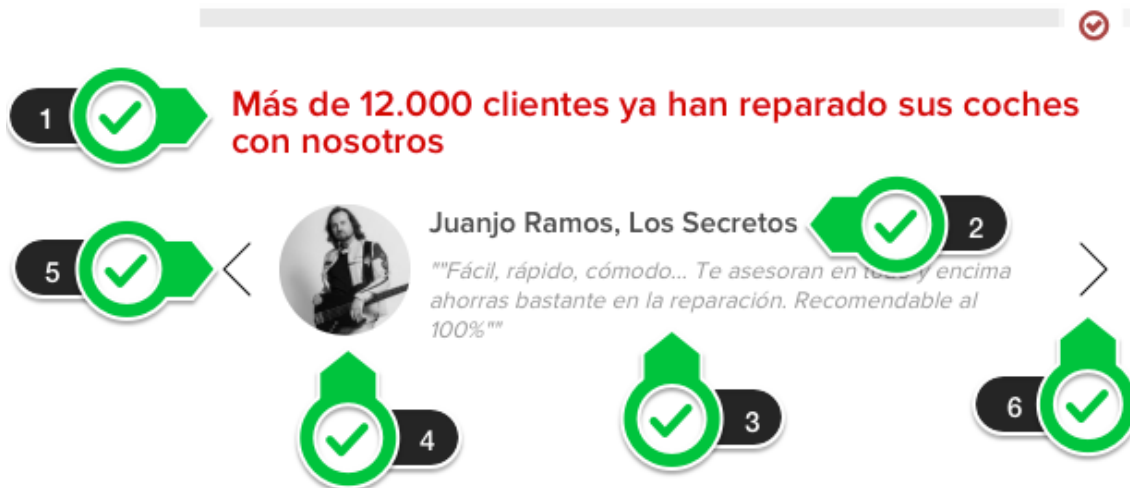


Ilustración 32. Testimonios de clientes (front-office)

1. Título para todo el módulo
2. Nombre de cliente
3. Testimonio del cliente
4. Fotografía del cliente
5. Testimonio anterior
6. Testimonio siguiente

La pantalla de gestión de este módulo permite crear un listado de testimonios (botón + testimonio). Cada testimonio se compone de un título (en este caso coincide con el nombre del cliente), una descripción, una foto, y un texto alternativo para la foto.

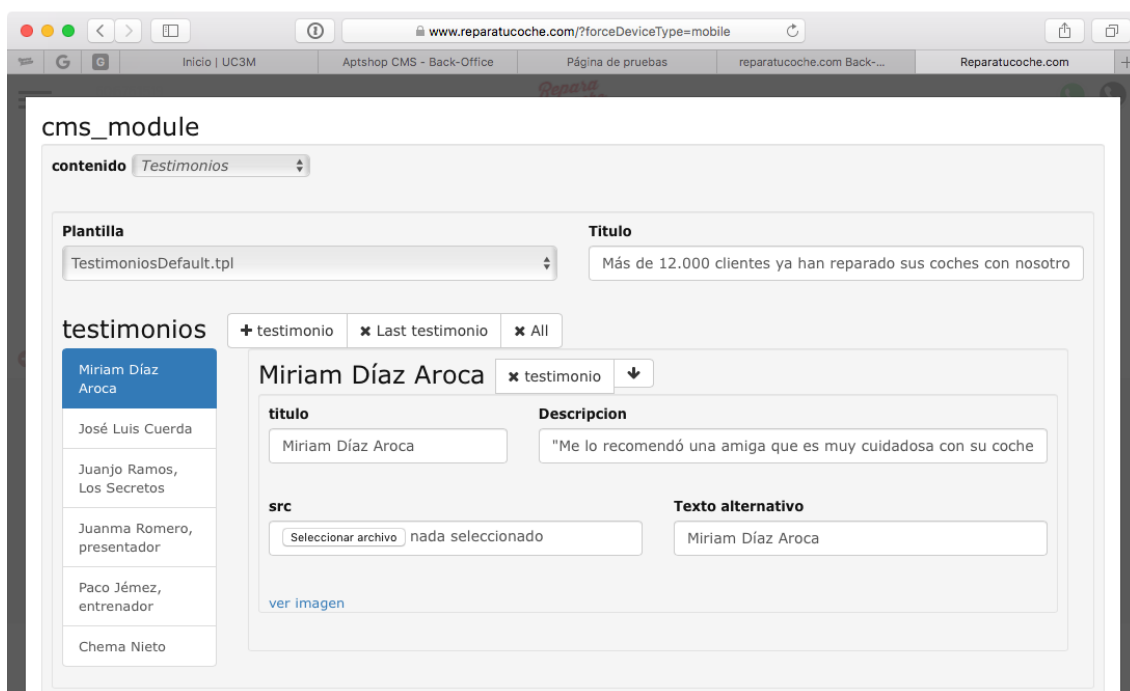


Ilustración 33. Testimonios de clientes (back-office)

5.7 Módulo de vídeo

El módulo de vídeo es muy sencillo de utilizar, al igual que fue muy sencillo de implementar. Se debe elegir primero al proveedor del servicio de video entre YouTube y Vimeo, se elige la proporción del vídeo (para conseguir una adaptación responsiva del mismo no se establece el tamaño) y se introduce el identificador del vídeo en su plataforma.



Ilustración 34. Módulo de vídeo (front-office)

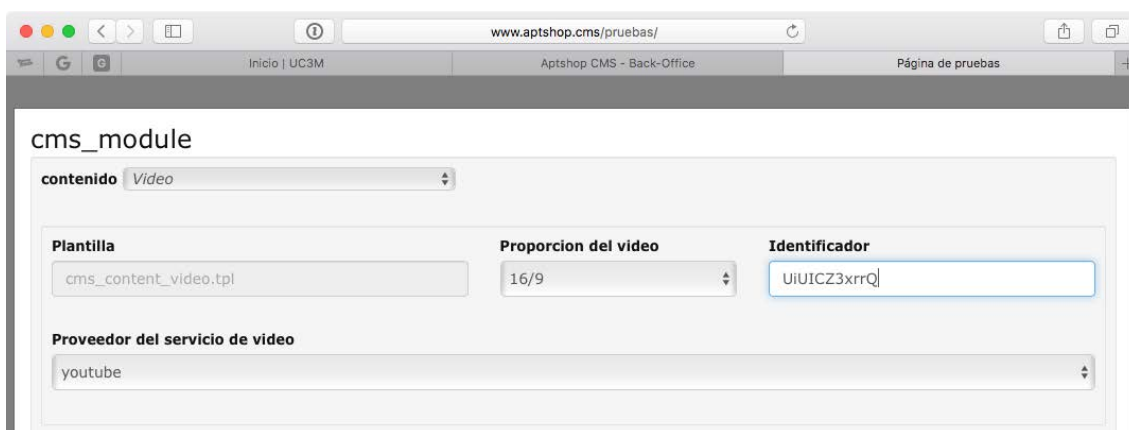


Ilustración 35. Módulo de vídeo (back-office)

5.8 Módulo de listado de productos

El módulo de listado de productos permite incrustar una selección de productos que cumplen unos criterios similares en un hueco o posición de una plantilla. Es un módulo muy versátil. Sus parámetros de configuración se pueden dividir en 2 puntos:

- **Parámetros de vista:** Nos permiten elegir como se verá el listado (si será carrusel o no, si estarán dispuestos en 1, 2 o 3 filas o en otro tipo de disposiciones, cuántos productos por fila queremos que se vean, y qué plantilla de producto queremos utilizar.
- **Parámetros de contenido:** Nos permiten seleccionar los productos en base a criterios de catálogo, como por ejemplo categorías y filtros. El número máximo de elementos a mostrar o si se tratan de novedades, superventas, ofertas o próximos lanzamientos.



Ilustración 36. Módulo de listado de productos (front-office)

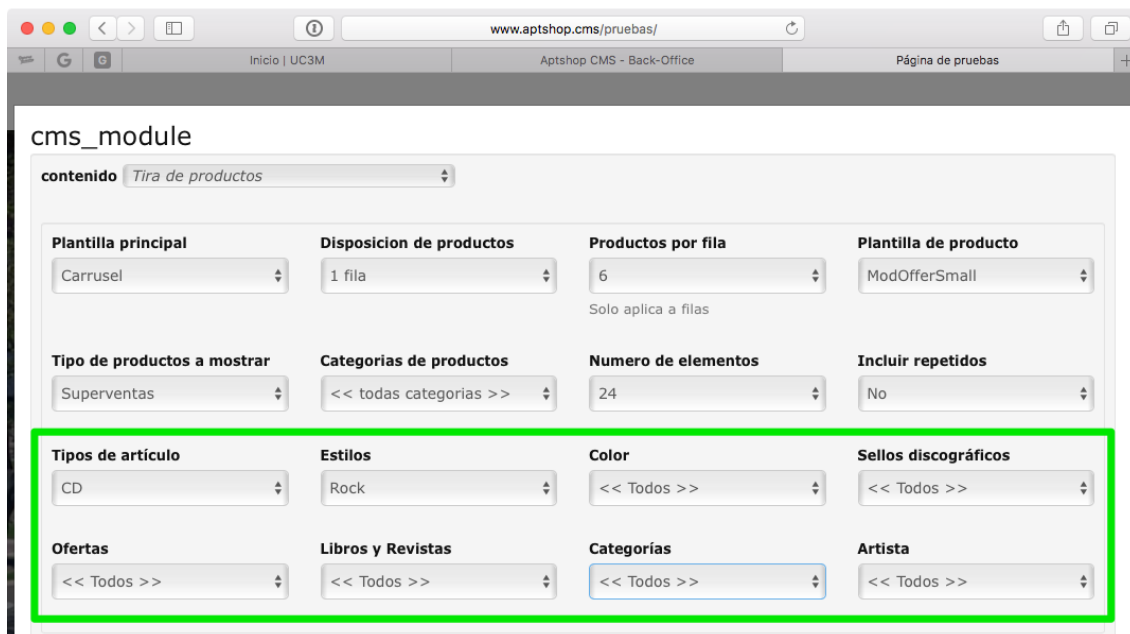


Ilustración 37. Módulo de listado de productos (back-office)

Los selectores que se encuentran dentro del recuadro verde son filtros de productos, los cuales se dan de alta y son distintos para cada una de las tiendas.

5.9 Módulo de título y descripción

Es un módulo muy sencillo que suele destacar por el diseño de la plantilla asociada. Se utiliza para que las páginas del sitio comiencen por el título y la descripción de la página, siendo estos parámetros clave para el SEO.



Ilustración 38. Módulo de título y descripción (front-office)

El módulo utiliza por defecto el título y la descripción de la página, pero estos pueden ser modificados en la pantalla de gestión del back-office. En esta pantalla, además, podemos seleccionar si invertir el color del texto y elegir los fondos que tendrá el módulo. Se pueden incluir un número indeterminado de fondos que, al tener la capacidad de ser transparentes, permiten construir fondos avanzados de manera sencilla y sin Photoshop. En el módulo del ejemplo, el fondo se compone de una imagen de un mecánico y de una imagen semitransparente (*overlay*) en degradado de transparente hacia gris oscuro, que hace que la tipografía en color blanco se lea sin problemas.

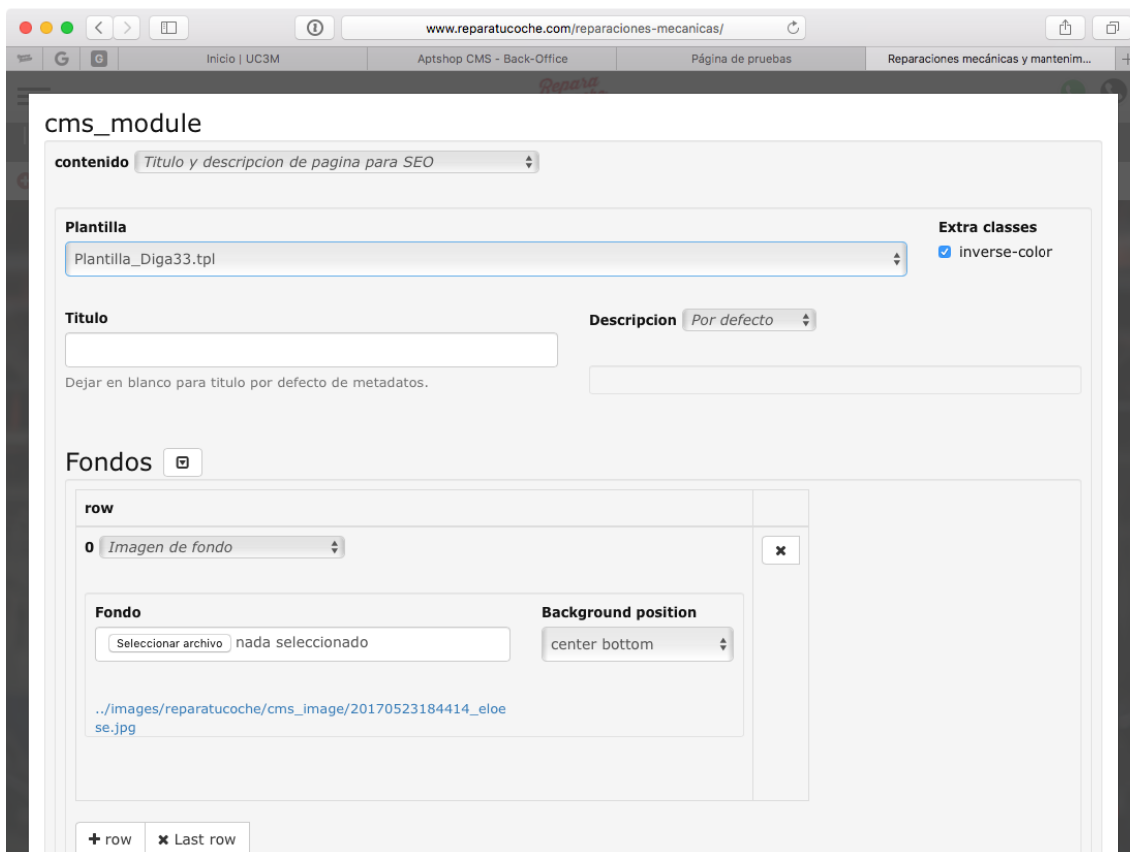


Ilustración 39. Módulo de título y descripción (back-office)

5.10 Estructura de columnas

Existen varios tipos de estructuras en columnas implementados en el CMS. La que vemos a continuación, junto con la pantalla de edición del back-office, es una estructura que: en ordenadores se verán cuatro columnas, pero que en tabletas se verán dos filas y dos columnas y que en móviles se verán 4 filas. Tiene un hueco, en el que se van añadiendo módulos, cada vez que se añade un módulo, aumenta el número de columnas, o da el salto a una nueva fila, según la configuración, el dispositivo de acceso y el número de elementos ya introducidos.

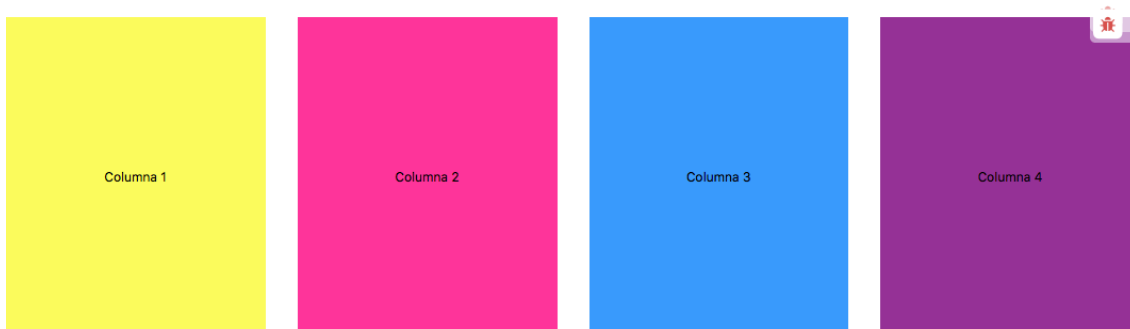


Ilustración 40. Estructura de columnas (front-office)

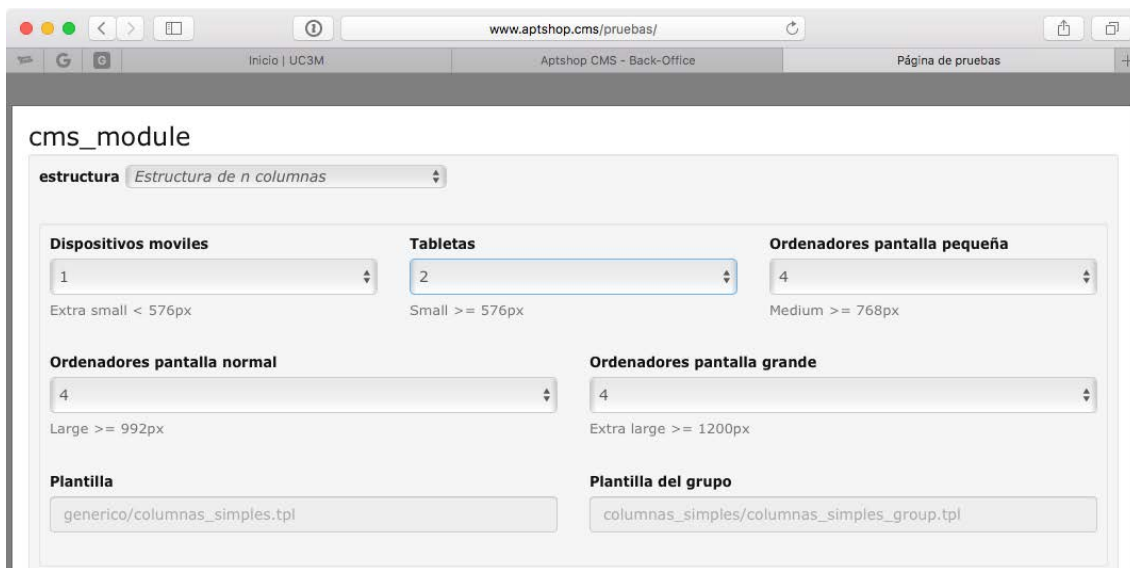


Ilustración 41. Estructura de columnas (back-office)

5.11 Estructura de puntos cardinales

Es una estructura típica para páginas web. Permite definir en un solo módulo una cabecera, un pie, una columna a la izquierda, una columna a la derecha y reservar en el centro un hueco más para el contenido principal. Esta estructura contiene un hueco por cada punto cardinal descrito. La configuración de esta estructura es mínima, permitiendo solamente elegir la plantilla.



Ilustración 42. Estructura de punto cardinales (front-office)

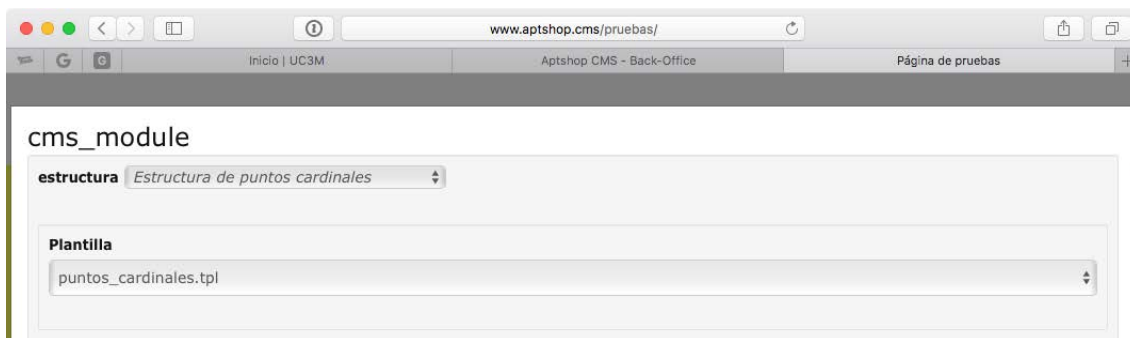


Ilustración 43. Estructura de puntos cardinales (back-office)

5.12 Estructura de pestañas

Al contrario que las estructuras anteriores, esta precisa la creación uno a uno de cada uno de los huecos. Dichos huecos tendrán asociados un nombre, que se utilizará como título de la pestaña.

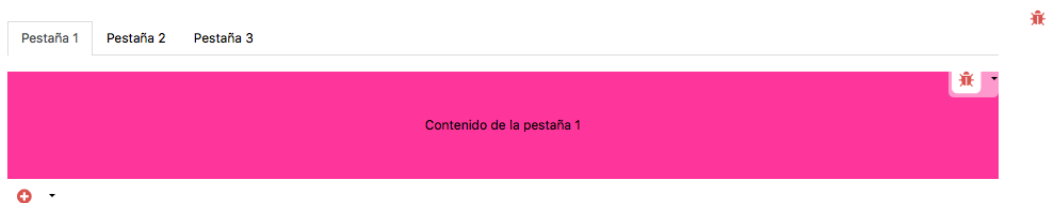


Ilustración 44. Estructura de pestañas (front-office)

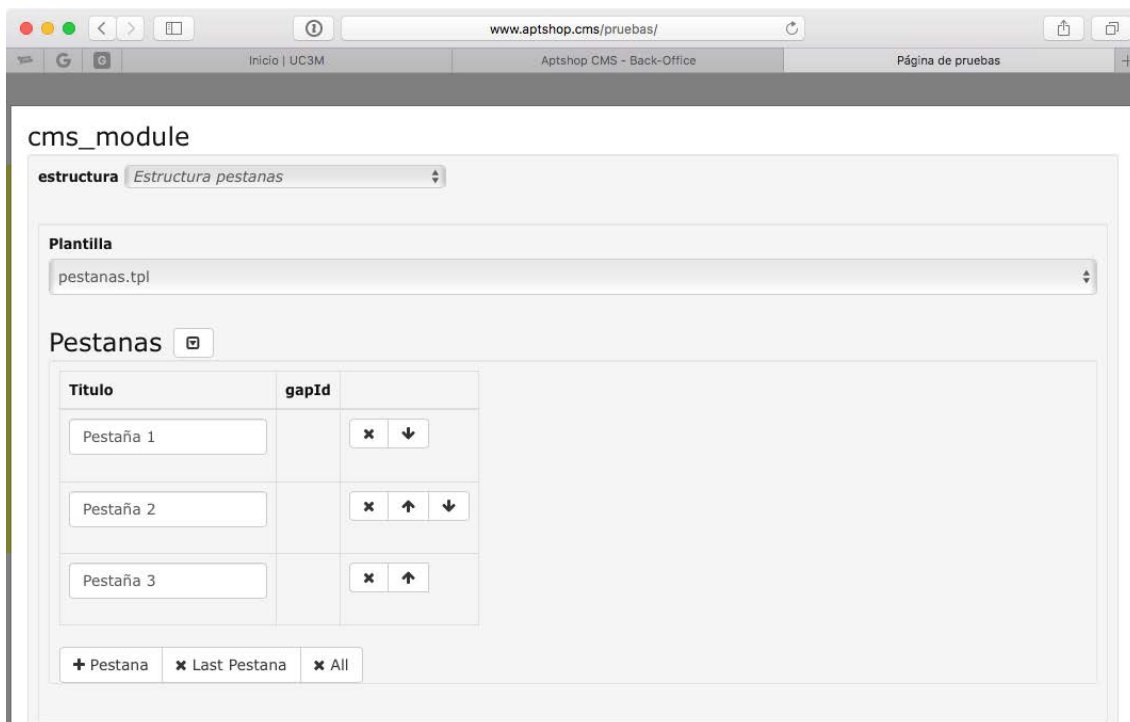


Ilustración 45. Estructura de pestañas (back-office)

5.13 Estructura de pase de diapositivas

Esta estructura tendrá un único hueco. Cada vez que se añada un módulo en el hueco, se creará una diapositiva más en el carrusel, la cual contendrá al módulo.



Ilustración 46. Estructura de pase de diapositivas (front-office)

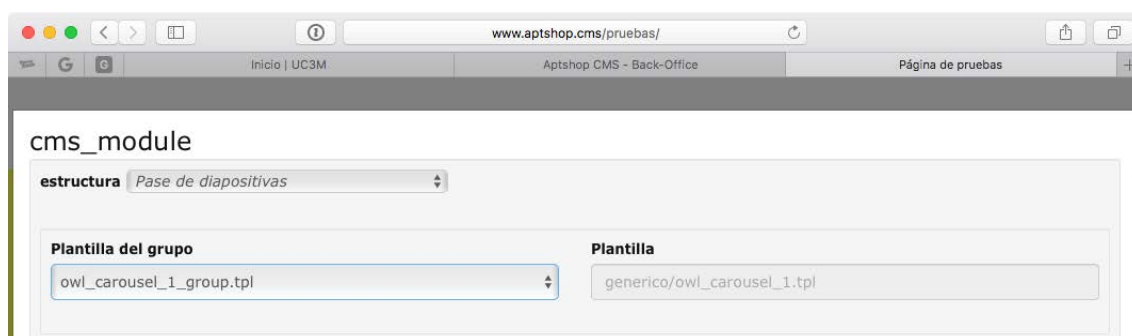


Ilustración 47. Estructura de pase de diapositivas (back-office)

5.14 Decoradores

La gestión de decoradores se ubica debajo de la gestión de módulos de CMS o de la de estructuras. Se pulsa sobre el botón “+ **Decorador**” para añadir un nuevo decorador al módulo.

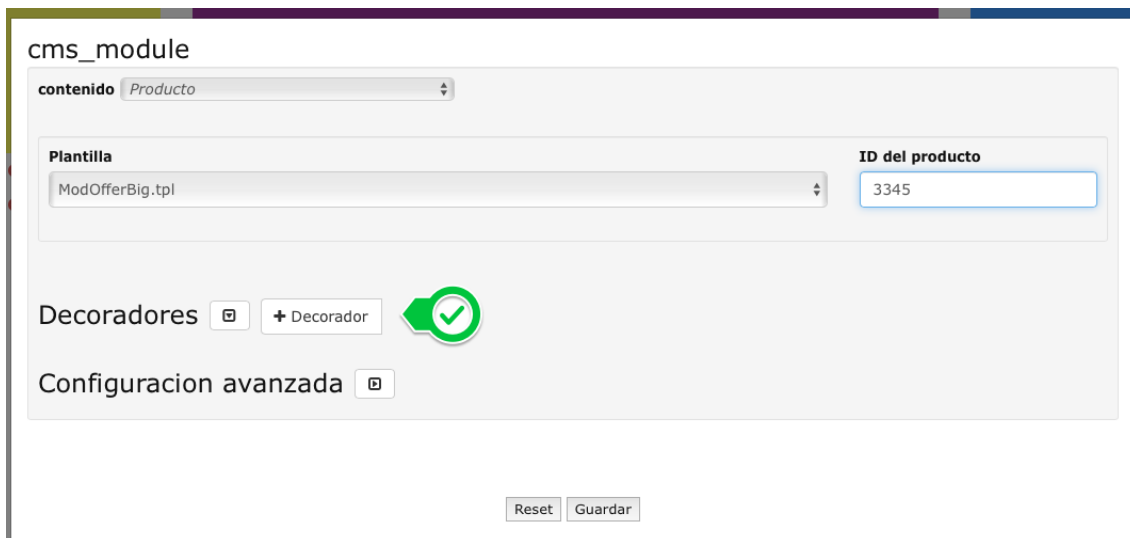


Ilustración 48. Gestión de decoradores



Ilustración 49. Gestión de decoradores. Decorador de título.

Al añadir un nuevo decorador (1), primero se deberá elegir el tipo de decorador (4) y aparecerán los campos de gestión del decorador seleccionado. Se podrán añadir tantos decoradores como se desee al módulo. Descripción de la imagen:

1. Añadir un nuevo decorador
2. Borrar último decorador
3. Borrar todos los decoradores
4. Elegir el tipo de decoradores
5. Los decoradores que se añadan al módulo aparecerán aquí.
6. Eliminar el decorador que se está visualizando.
7. Botones de ordenar los distintos decoradores.

El decorador de título añadido sobre una estructura de pase de diapositivas se vería así:

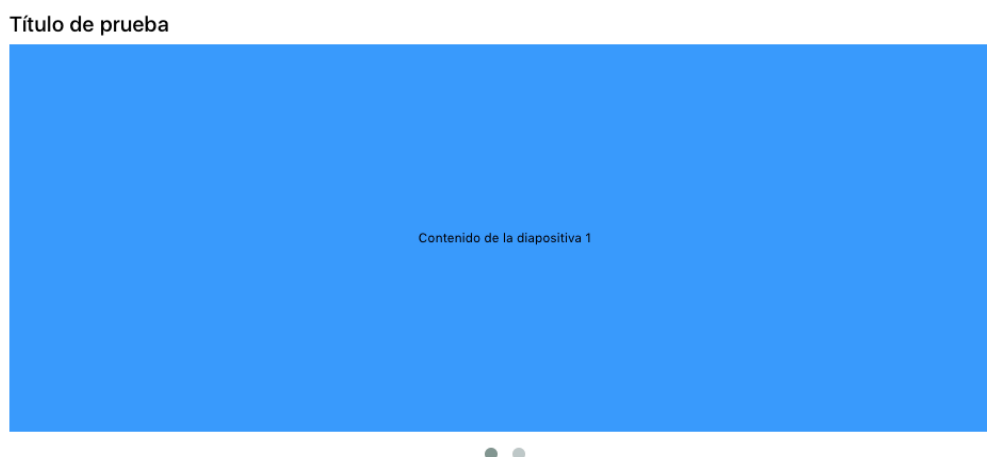


Ilustración 50. Decorador de título (front-office)

Otros decoradores:

Un decorador de enlace permite que al pulsar en un módulo, este esté enlazado todo él con un destino. Este decorador sólo es compatible con HTML5 y el módulo en cuestión no debe tener enlaces dentro. Esta máxima no se comprueba, por lo que el editor debe ser consciente de esta restricción, por lo que se ha puesto un comentario en el panel de gestión.

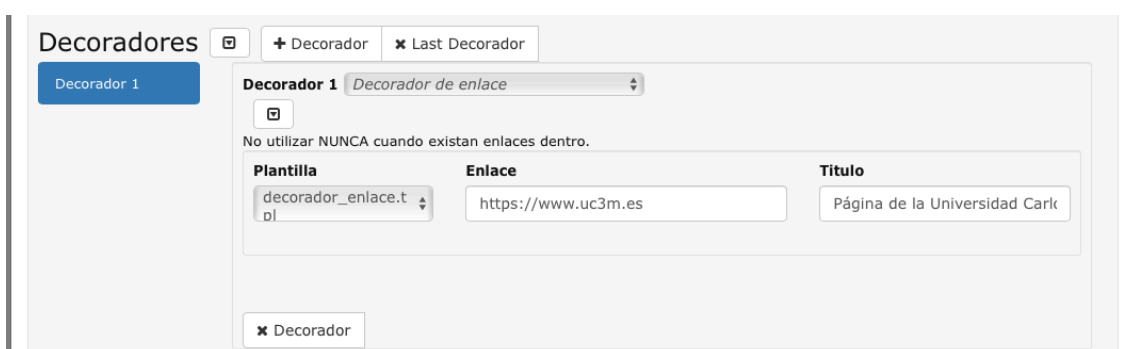


Ilustración 51. Decorador de enlace (back-office)

El decorador de bloque es muy versátil. Permite seleccionar color de texto, color de fondo, borde, margen, *padding*. Permite añadir clases de HTML a usuarios avanzados, o incluso definir el estilo de la etiqueta.

Decoradores

Decorador 1

+ Decorador

Last Decorador

Decorador 1

Decorador bloque

Coloca este decorador el ultimo para que incluya al resto de decoradores y queden dentro.

Plantilla

decorador_bloque.tpl

Color

primary

Margen interior (padding)

izquierda_derecha

Margen exterior (margin)

superior_inferior

Borde

sin_borde

Estra classes

text-center

Style

Decorador

Ilustración 52. Decorador de bloque (back-office)

6 Gestión del proyecto

6.1 Introducción

En este apartado se describirán los recursos que se utilizarán, tanto hardware, como software o recursos humanos. Se planificará la ejecución del proyecto

Aptitude Web Services, la empresa que desarrolla este software es una micro empresa, en la que trabajan 3 personas. Tiene oficina en San Sebastián de los Reyes y dispone de equipos informáticos para cada empleado, así como infraestructura tecnológica en la nube. Hace un gran uso de software libre, ya que la mayor parte del software utilizado es bajo este tipo de licencias.

6.2 Estimación de recursos

6.2.1 Recursos hardware

Se utilizará el material interno de la empresa. Los equipos informáticos de dos de los perfiles ya se encuentran amortizados. El tercero tiene un coste de amortización de 1,33€ al día.

6.2.2 Recursos software

Los empleados de Aptitude Web Services utilizan sistemas operativos Ubuntu Linux o Mac. Para el desarrollo de aplicaciones web PHP, JavaScript, CSS y MySQL utilizan NetBeans. Para este proyecto en particular se ha contado con una licencia de Microsoft Office 365 para la realización de la documentación.

En cuanto a servidores de desarrollo, se utiliza Subversion en una máquina Ubuntu Linux en la nube de Amazon. Esta máquina es compartida con el resto de proyectos. Y otra máquina más para pruebas en preproducción.

6.2.3 Recursos humanos

Fernando Serrano Elvira, CEO de la empresa y programador con amplia experiencia. En el momento del desarrollo del CMS su tiempo es muy limitado. Por lo que no se podrá contar casi con su participación en el proyecto.

Daniel Peña Gil, CTO de la empresa, ingeniero en informática a falta del proyecto final de carrera con 7 años de experiencia profesional. Domina las arquitecturas de software de los CMS y está familiarizado con su uso. Tiene experiencia tanto en la gestión de proyectos de desarrollo de software, como de programador en tecnologías PHP, JavaScript, CSS y SQL. Al ser socio de la empresa y al tratarse de un proyecto interno tiene un coste para la misma de 18€/hora. Disponibilidad total para el proyecto.

Empleada desarrolladora, graduada en informática con 3 años de experiencia en el desarrollo de la plataforma Aptshop. Tiene experiencia en desarrollo con las tecnologías PHP, JavaScript, CSS y SQL. Un coste para la empresa de 14,20€/hora. Disponibilidad limitada a 5h/semana.

6.3 Plantificación

Id.	Nombre de tarea	T	Inicio	Fin	H	Recursos
1	Análisis del sistema	41d	30-01-17	01-03-17	328h	
2	Captura de requisitos de usuario	5d	30-01-17	03-02-17	40h	Daniel
3	Definición de casos de uso	10d	06-02-17	17-02-17	80h	Daniel
4	Definición de requisitos de software	5d	20-02-17	24-02-17	40h	Daniel
5	Documentación de los requisitos	3d	27-02-17	01-03-17	24h	Daniel
6	Diseño de la arquitectura de front-office	9d	02-03-17	14-03-17	72h	Daniel
7	Diseño del modelo	5d	02-03-17	08-03-17	40h	Daniel
8	Diseño del Controlador	2d	09-03-17	10-03-17	16h	Daniel
9	Diseño de la interfaz de gestión	2d	13-03-17	14-03-17	16h	Daniel
10	Diseño de la arquitectura de back-office	20d	15-03-17	28-03-17	100h	
11	Gestión de paginas	10d	15-03-17	28-03-17	20h	Progr.
12	Gestión de módulos	2d	15-03-17	16-03-17	16h	Daniel
13	Gestión de posicionamiento de los módulos	2d	17-03-17	20-03-17	16h	Daniel
14	Gestión de decoradores	2d	21-03-17	22-03-17	16h	Daniel
15	Diseño de extensores de esquemas JSON	4d	23-03-17	28-03-17	32h	Daniel
16	Implementación	89d	29-03-17	31-05-17	436h	
17	Implementación del modelo de base de datos	8d	29-03-17	07-04-17	16h	Progr.
18	Carga de información de prueba	2d	10-04-17	11-04-17	4h	Progr.
19	Integración de JSON Editor en scripts de back-office	10d	29-03-17	11-04-17	80h	Daniel
20	Acceso al modo edición	8d	12-04-17	21-04-17	16h	Progr.
21	Control de carga módulos y decoradores front-office	5d	12-04-17	18-04-17	40h	Daniel
22	Control de carga de paginas de CMS front	1d	19-04-17	19-04-17	8h	Daniel
23	Maquetación de 16 módulos CMS	8d	20-04-17	01-05-17	64h	Daniel
24	Programación JavaScript 5 módulos CMS	28d	24-04-17	31-05-17	56h	Progr.
25	Implementación de la interfaz de edición	9d	02-05-17	17-05-17	72h	Daniel
26	Implementación gestión de módulos CMS en back	10d	18-05-17	31-05-17	80h	Daniel
27	Pruebas	25d	01-06-17	05-07-17	250h	
28	Pruebas de front-office	10d	01-06-17	14-06-17	100h	P,D
29	Pruebas de back-office	10d	15-06-17	28-06-17	100h	P,D
30	Pruebas de integración	5d	29-06-17	05-07-17	50h	P,D

Tabla 85. Planificación de tareas

6.3.1 Diagrama de Gantt

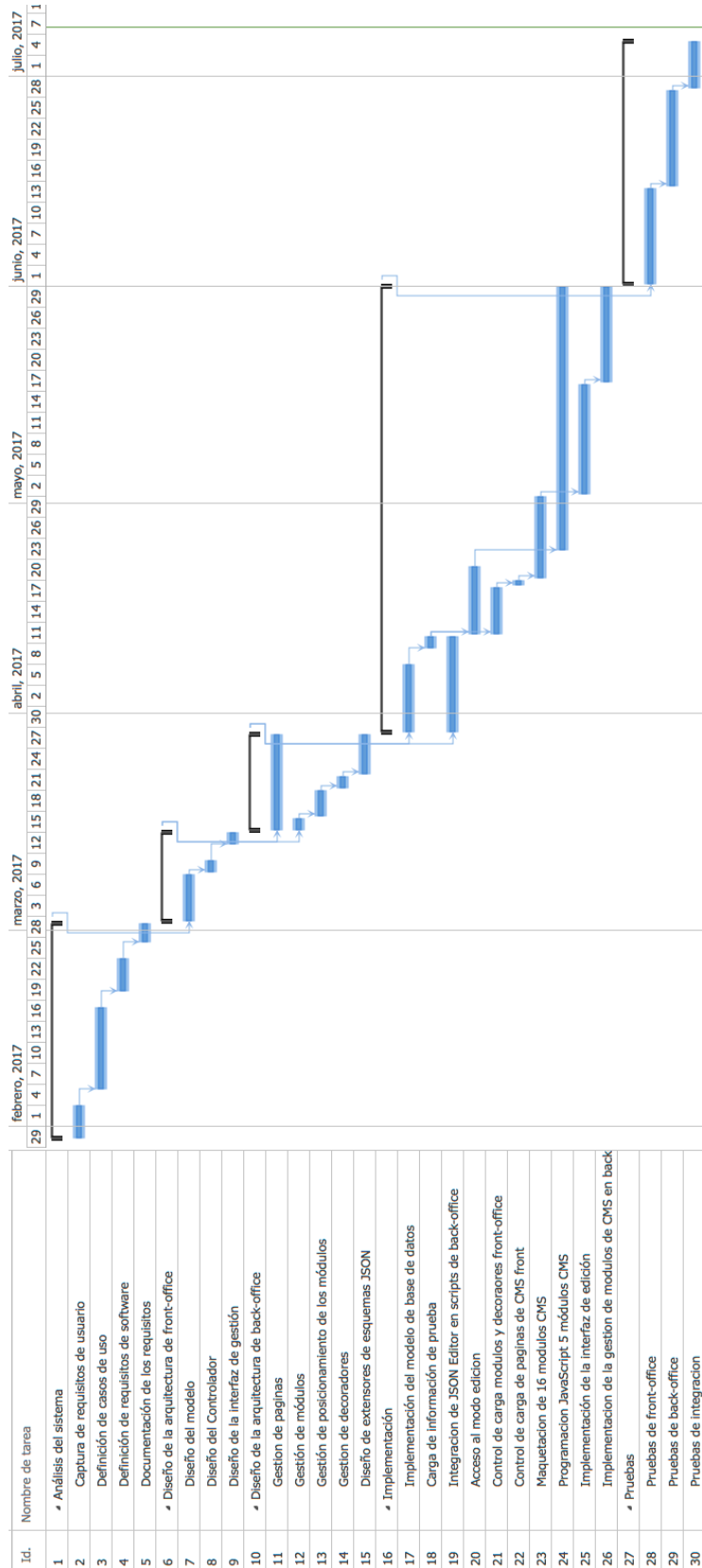


Ilustración 53. Diagrama de Gantt

6.4 Presupuesto

En este apartado se desglosarán los costes directos que se generan por la creación del proyecto, así como una proporción de los gastos de empresa atribuibles indirectamente al mismo.

6.4.1 Costes directos

Concepto	Unidades	Precio unidad (€)	Atribuible	Total (€)
Daniel	109 días	144,00	100%	15.696,00
Programador	73 días	113,60	25%	2.073,20
Ordenador 1	109 días	0,00 ¹⁶	100%	0,00
Ordenador 2	73 días	1,33	25%	24,27
Microsoft Office	6 meses	10,70	100%	64,20
Total				17.857,67

Tabla 86. Costes directos

6.4.2 Costes indirectos

Concepto	Unidades	Precio unidad (€)	Atribuible	Total (€)
Oficina	109 días	18,50	35%	705,78
Internet + telf.	109 días	2,00	35%	76,30
EC2¹⁷	109 días	0,44	35%	16,79
Subversion				
EC2 pruebas + integración	25 días	0,44	100%	11,00
Total				809,87

Tabla 87. Costes indirectos

¹⁶ Como vimos en el punto anterior, este equipo tiene ya 6 años y se encuentra amortizado.

¹⁷ EC2 es el servicio de computación de Amazon en la nube.

6.4.3 Costes totales

Concepto	Coste	Costes inesperados (10%)	Total (€)
Costes directos	17.857,67	1.785,77	19.643,44
Costes indirectos	809,87	80,99	890,86
Total			20.534,30

Tabla 88. Costes totales

7 Conclusiones y trabajos futuros

7.1 Resultados obtenidos

Se ha desarrollado un *plugin* para Aptshop que permite a usuarios no técnicos, modificar prácticamente cualquier apartado de contenido de una tienda Aptshop. El resultado ha sido tan satisfactorio que este *plugin* se incluye en todas las instalaciones de Aptshop desde que se desarrolló. Además, uno de los criterios que se tuvo en cuenta a la hora de desarrollarlo, es que integrar nuevos módulos o plantillas se hiciera de manera muy ágil. Esto nos permite una reducción de costes en determinadas tareas que justifican su inversión.

7.1.1 Estandarización en el desarrollo de front-office

Ahora existe una metodología clara en el desarrollo de módulos de contenido para el front-office. Unificando las metodologías, nos resulta más sencillo que distintos programadores entiendan y colaboren en un mismo código.

7.1.2 Mejoras en la calidad de servicio

Antes de existir el CMS, a la hora de integrar una nueva plantilla, los módulos relativos a comercio electrónico eran dinámicos, no así el resto de contenido (salvo algunas excepciones). Con el nuevo CMS, y con un mínimo de esfuerzo extra (definir el esquema de JSON del módulo), entregamos una tienda con el contenido no de tienda dinámico, permitiendo a nuestros clientes realizar modificaciones.

7.1.3 Mejoras en tiempos de desarrollo

Los pocos módulos dinámicos no de tienda que existían, tuvieron un coste de desarrollo muy alto, reduciéndose el coste a una décima parte después de la implantación del CMS. Por ejemplo, el módulo de *banners* dinámicos preexistente tuvo un coste de desarrollo de en torno a una semana. El nuevo módulo de *banners* hecho con el CMS se pudo desarrollar en apenas unas horas.

7.1.4 Eficiencia en proyectos actuales

En algunos proyectos, nos ha permitido delegar en nuestros clientes labores de modificaciones de contenidos que antes eran hechas por nuestros desarrolladores sin la modificación de las condiciones del contrato. Esto nos supone un ahorro, sólo en proyectos actuales de casi 5.000 € anuales.

En otros proyectos en lo que seguimos asumiendo la gestión de contenidos nosotros, sumando el tiempo de desarrollo de módulos de CMS y restando el tiempo de modificación de los contenidos y de la eliminación de fallos, ha permitido reducir el tiempo de gestión de los contenidos considerablemente en el medio plazo. Calculamos

unos ahorros en costes de 3.000 € en los próximos 12 meses. Y de 8.000 € más, en los siguientes 12, ya que los módulos de CMS estarán ya implementados.

7.1.5 Nuevos proyectos

El disponer del CMS nos ha permitido obtener nuevos proyectos con unas características distintas, en las cuales los clientes dependen menos de nosotros, y supone un ahorro de costes tanto para ellos como para nosotros. En concreto, con este modelo, se ha comenzado a trabajar en dos nuevos proyectos, uno de tienda, en el que esperamos obtener un beneficio estimado de 16.000 € en dos años y otro únicamente de contenidos con un beneficio de en torno a 2.000 €.

7.1.6 Retorno de la inversión

Por lo tanto, esperamos recuperar el 100% de la inversión entre el primer y el segundo año después de su puesta en marcha.

7.2 Dificultades encontradas

Desde el principio se supo que este proyecto iba a suponer un coste bastante alto para una pequeña empresa como la nuestra. Por lo tanto, se buscó por todos los medios la integración de cualquier sistema ya desarrollado dentro de nuestra plataforma. Viendo lo complicado que se hacía, se optó por el desarrollo propio con el resultado expuesto en este documento.

Siendo uno de los objetivos que el desarrollo de nuevos módulos tuviera un coste bajo, supuso un desafío importante el desarrollo de la arquitectura de back-office. Como ya hemos comentado, se solucionó con la incorporación de un modelo basado en JSON Schema y el software JSON Editor de Jeremy Dorn actuando como generador de vistas. Esto nos permitió centrarnos mayoritariamente en el desarrollo del controlador.

7.3 Líneas futuras

El CMS tiene unas vías de desarrollo intrínsecas a la vida de la empresa. Según se vayan desarrollando nuevos proyectos, todos los nuevos módulos se incorporarán a Aptshop, de modo que el número de los mismos no va a parar de crecer. En algunas ocasiones, lo que se hará será modificar los módulos antiguos para incluirles nueva funcionalidad, intentando en todo momento no complicar en exceso su uso.

Por otro lado, se establece como punto importante, el mejorar la experiencia de usuario en la utilización del CMS. Permitir opciones ahora no implementadas como copiar, cortar y pegar módulos o desplazarlos mediante un sistema de arrastrar y soltar serían muy deseables.

El mejorar estos puntos nos permite valorar el desarrollo de una nueva vía de negocio en la que el software se contrate de modo online y sin intervención humana, aunque este punto requiere un estudio de viabilidad más exhaustivo.

8 Bibliografía

- Wikipedia, «Sistema de gestión de contenidos,» 2017. [En línea].
- 1] Available: https://es.wikipedia.org/wiki/Sistema_de_gestión_de_contenidos. [Último acceso: 2017].
- W3Techs.com, «Usage of content management systems for websites,» 6 5
- 2] 2017. [En línea]. Available: https://w3techs.com/technologies/overview/content_management/all. [Último acceso: 6 5 2017].
- Wordpress, «WordPress Now Available,» 27 05 2003. [En línea].
- 3] Available: <https://wordpress.org/news/2003/05/wordpress-now-available/>. [Último acceso: 06 05 2017].
- BuiltWith, «Joomla! usage statistics and trends,» 06 05 2017. [En línea].
- 4] Available: <https://trends.builtwith.com/cms/Joomla!>. [Último acceso: 06 05 2017].
- Joomla!, «Historia de la Versión - Joomla Documentation,» 26 08 2015.
- 5] [En línea]. Available: https://docs.joomla.org/Category:Version_History/es. [Último acceso: 07 05 2017].
- Wikipedia, «Mambo (software),» 01 04 2017. [En línea]. Available:
- 6] [https://en.wikipedia.org/w/index.php?title=Mambo_\(software\)&oldid=773236869](https://en.wikipedia.org/w/index.php?title=Mambo_(software)&oldid=773236869). [Último acceso: 06 05 2017].
- Open Source Matters, Inc., «Extension types (general definitions),» 08 01
- 7] 2015. [En línea]. Available: [https://docs.joomla.org/Extension_types_\(general_definitions\)](https://docs.joomla.org/Extension_types_(general_definitions)). [Último acceso: 06 04 2017].
- Drupal, «Our history,» 28 10 2015. [En línea]. Available:
- 8] <https://www.drupal.org/about/history>. [Último acceso: 07 05 2017].
- Wikipedia, «Magento,» 04 05 2017. [En línea]. Available:
- 9] <https://en.wikipedia.org/w/index.php?title=Magento&oldid=778701940>. [Último acceso: 07 05 2017].
- Wikipedia, «PrestaShop,» 05 05 2017. [En línea]. Available:
- 10] <https://es.wikipedia.org/wiki/PrestaShop>. [Último acceso: 10 05 2017].
- PrestaShop, «E-commerce Industry Partners - PrestaShop,» 08 05 2017. [En

11] línea]. Available: <https://www.prestashop.com/en/industry-partners>. [Último acceso: 08 05 2017].

Z. Ait-Bahajji y G. Illyes, «HTTPS as a ranking signal,» 06 08 2014. [En

12] línea]. Available: <https://webmasters.googleblog.com/2014/08/https-as-ranking-signal.html>. [Último acceso: 02 06 2017].

K. E. Wiegers y J. Beatty, Software Requirements, 3ª edición ed., Microsoft

13] Press, 2013.

C. Larman, UML y patrones: una introducción al análisis y diseño orientado

14] a objetos y al proceso unificado, Pearson Educación, 2003, p. 590.

M. A. Álvarez Pacheco, «“Desarrollo del Portal Web de la Universidad

15] Nacional de Loja, aplicando técnicas de web semántica en el gestor de contenidos Drupal,» Loja, 2014.

R. Hamilton y K. Miles, Learning UML 2.0, O'Reilly Media, 2006.

16]

9 Anexo I: JSON Schema de los módulos de CMS.

9.1 Módulo *Banners*

```
{
  "title": "Slideshow de Banners",
  "type": "object",
  "properties": {
    "media_types_name": {
      "title": "Tipo de banner",
      "type": "string",
      "enum": [
        "banners_image_small",
        "banners_image_big",
        "banners_image_news"
      ],
      "propertyOrder": 10
    },
    "mode": {
      "title": "Modo de funcionamiento",
      "oneOf": [
        {
          "title": "Por defecto",
          "type": "object",
          "properties": {
            "type": {
              "title": "modo",
              "type": "string",
              "template": "default"
            }
          }
        },
        {
          "title": "Personalizado",
          "type": "object",
          "properties": {
            "type": {
              "title": "modo",
              "type": "string",
              "template": "custom"
            }
          }
        }
      ],
      "options": {
        "collapsed": false,
        "disable_collapse": true,
        "disable_edit_json": true,
        "disable_properties": true
      }
    }
  }
}
```

9.2 Módulo *Producto*

```
{
  "title": "Producto",
  "type": "object",
  "properties": {
    "template": {
      "title": "Plantilla",
      "type": "string",
      "enum": [
        "ModOfferBig.tpl",
        "ModOfferInfoProducto.tpl",
        "ModOfferError.tpl",
        "ModOfferSmall.tpl",
        "ModOfferImageCarouselBig.tpl",
        "ModOfferFiltersList.tpl",
        "ModOfferImageAndSocial.tpl"
      ],
      "propertyOrder": 2
    },
    "products_id": {
      "title": "ID del producto",
      "type": "integer",
      "propertyOrder": 3
    }
  }
}
```

```
}
}
```

9.3 Módulo *Testimonio* *Clientes*

```
{
  "title": "Testimonios",
  "type": "object",
  "properties": {
    "template": {
      "options": {
        "templatesDir": "Testimonios",
        "grid_columns": 6
      },
      "propertyOrder": 10
    },
    "title": {
      "title": "Titulo",
      "type": "string",
      "propertyOrder": 20,
      "options": {
        "grid_columns": 6
      }
    },
    "testimonios": {
      "type": "array",
      "format": "tabs",
      "title": "testimonios",
      "uniqueItems": false,
      "items": {
        "type": "object",
        "title": "testimonio",
        "headerTemplate":
          "{{self.title}}",
        "properties": {
          "title": {
            "title": "titulo",
            "type": "string",
            "propertyOrder": 10,
            "options": {
              "grid_columns": 4
            }
          },
          "description": {
            "title": "Descripcion",
            "type": "string",
            "options": {
              "grid_columns": 8
            }
          },
          "propertyOrder": 20
        },
        "src": {
          "type": "string",
          "format": "url",
          "options": {
            "upload": true,
            "grid_columns": 6
          }
        },
        "links": [
          {
            "href": "../{{self}}",
            "rel": "ver imagen"
          }
        ],
        "alt": {
          "title": "Texto alternativo",
          "type": "string",
          "options": {
            "grid_columns": 6
          }
        }
      }
    }
  }
}
```

```

    },
    "options": {
      "collapsed": false,
      "disable_collapse": true,
      "disable_edit_json": true,
      "disable_properties": true,
      "grid_columns": 12
    }
  },
  "options": {
    "collapsed": false,
    "disable_collapse": true,
    "disable_edit_json": true,
    "disable_properties": true,
    "grid_columns": 12
  },
  "propertyOrder": 40
}
}
}

```

9.4 Módulo Imágenes

```

{
  "title": "Imagen",
  "type": "object",
  "properties": {
    "template": {
      "title": "Plantilla",
      "type": "string",
      "enum": [
        "image.tpl",
        "image_with_popover.tpl"
      ],
      "propertyOrder": 10
    },
    "src": {
      "type": "string",
      "format": "url",
      "options": {
        "upload": true
      },
      "links": [
        {
          "href": "../{{self}}",
          "rel": "ver imagen"
        }
      ]
    },
    "alt": {
      "title": "Texto alternativo",
      "type": "string"
    },
    "class": {
      "title": "Class",
      "type": "string",
      "enum": [
        "img-responsive",
        "img-responsive center-block",
        ""
      ]
    }
  }
}
}

```

9.5 Módulo Vídeo

```

{
  "title": "Video",
  "type": "object",
  "properties": {
    "template": {
      "title": "Plantilla",
      "type": "string",
      "template": "cms_content_video.tpl",
      "propertyOrder": 1
    },
    "proporcion": {

```

```

      "title": "Proporcion del video",
      "type": "string",
      "enum": [
        "16/9",
        "4/3"
      ],
      "propertyOrder": 2
    },
    "identificador": {
      "title": "Identificador",
      "type": "string",
      "propertyOrder": 3
    },
    "servicio": {
      "title": "Proveedor del servicio de video",
      "type": "string",
      "enum": [
        "youtube",
        "vimeo"
      ],
      "propertyOrder": 4
    }
  }
}

```

9.6 Módulo Listado de Productos

```

{
  "title": "Tira de productos",
  "type": "object",
  "properties": {
    "template": {
      "title": "Plantilla principal",
      "type": "string",
      "enum": [
        "default",
        "ModFeaturedProductsDisenoEstatico.tpl"
      ],
      "options": {
        "enum_titles": [
          "Carrusel",
          "Sin Carrusel"
        ]
      },
      "grid_columns": 3
    },
    "propertyOrder": 2
  },
  "rows": {
    "title": "Disposicion de productos",
    "type": "string",
    "enum": [
      "ModFeaturedProducts_group1.tpl",
      "ModFeaturedProducts_group2.tpl",
      "ModFeaturedProducts_group3.tpl",
      "ModFeaturedProductsDisenoCuadrícula.tpl"
    ]
  },
  "options": {
    "enum_titles": [
      "1 fila",
      "2 filas",
      "3 filas",
      "1 p grande + 8 pequenos"
    ]
  },
  "grid_columns": 3
},
"propertyOrder": 5
},
"numeroitems": {
  "title": "Productos por fila",

```

```

        "description": "Solo aplica a
filas",
        "type": "string",
        "enum": [
            "1",
            "2",
            "3",
            "4",
            "5",
            "6"
        ],
        "options": {
            "grid_columns": 3
        },
        "propertyOrder": 8
    },
    "featuredType": {
        "title": "Tipo de productos a
mostrar",
        "type": "string",
        "enum": [
            "SITE_FILTER_NOVEDADES",
            "SITE_FILTER_SUPERVENTAS",
            "SITE_FILTER_SUPERVENTAS_HOME_TIPO",
            "SITE_FILTER_OFERTAS",
            "SITE_FILTER_PROXIMOS LANZAMIENTOS",
            "USER_RECENTLY_VIEWED_PRODUCTS",
            "PRODUCT_RELATED_PRODUCTS",
            "PRODUCTS_LIST"
        ],
        "options": {
            "enum_titles": [
                "Novedades",
                "Superventas",
                "Superventas home tipo",
                "Ofertas",
                "Proximos lanzamientos",
                "Vistos recientemente",
                "Productos relacionados",
                "Listado de productos"
            ],
            "grid_columns": 3
        },
        "propertyOrder": 10
    },
    "categoriesFilter": {},
    "modOfferTemplate": {
        "title": "Plantilla de producto",
        "type": "string",
        "enum": [
            "ModOfferSmall.tpl",
            "ModOfferSmallWithoutText.tpl",
            "ModOfferSmallWithTitle.tpl"
        ],
        "options": {
            "enum_titles": [
                "ModOfferSmall",
                "ModOfferSmall sin texto",
                "ModOfferSmall con titulo"
            ],
            "grid_columns": 3
        },
        "propertyOrder": 9
    },
    "numberOfElements": {
        "title": "Numero de elementos",
        "type": "string",
        "enum": [
            "1",
            "2",
            "3",
            "4",
            "5",
            "6",
            "8",
            "10",

```

```

            "12",
            "24",
            "36",
            "48"
        ],
        "options": {
            "grid_columns": 3
        },
        "propertyOrder": 60
    },
    "includeRepeatedMode": {
        "title": "Incluir repetidos",
        "type": "string",
        "default": "0",
        "enum": [
            "0",
            "1",
            "2"
        ],
        "options": {
            "enum_titles": [
                "No",
                "Si",
                "No nunca"
            ],
            "grid_columns": 3
        },
        "propertyOrder": 66
    }
}

```

9.7 Módulo Plantillas Predefinidas

```

{
    "title": "Plantilla predefinida",
    "type": "object",
    "properties": {
        "template": {
            "options": {
                "templatesDir":
"predefinedTemplates"
            },
            "propertyOrder": 2
        }
    }
}

```

9.8 Módulo Formulario Captación

```

{
    "title": "Formulario de registro",
    "type": "object",
    "properties": {
        "template": {
            "title": "Plantilla",
            "type": "string",
            "enum": [
                "formulario_registro_simple.tpl",
                "formulario_registro_home.tpl"
            ],
            "propertyOrder": 10
        },
        "origin_form_name": {
            "title": "Nombre del formulario",
            "description": "Se puede usar para
saber en que formulario se ha dado de
alta el usuario (estadisticas) ",
            "type": "string",
            "propertyOrder": 20
        },
        "action": {
            "title": "Accion",

```

```

        "type": "string",
        "enum": [
            "create_account",
            "create_power_account"
        ],
        "propertyOrder": 30
    },
    "autologin": {
        "title": "Autologin",
        "type": "string",
        "enum": [
            "true",
            "false"
        ],
        "propertyOrder": 40
    },
    "redirect_url": {
        "title": "URL al completar
registro",
        "description": "Dejar en blanco
para recargar pagina. ",
        "type": "string",
        "propertyOrder": 50
    },
    "conditions_texts_and_links": {
        "title": "Texto y enlaces
condiciones y privacidad",
        "description": "Dejar en blanco
para condiciones por defecto. ",
        "type": "string",
        "propertyOrder": 60
    }
}
}

```

9.9 Módulo Título y Descripción

```

{
    "title": "Titulo y descripcion de
pagina para SEO",
    "type": "object",
    "properties": {
        "template": {
            "options": {
                "templatesDir":
"ModCmsSeoPageTitle"
            },
            "propertyOrder": 10
        },
        "class": {
            "title": "Extra classes",
            "type": "array",
            "uniqueItems": true,
            "items": {
                "type": "string",
                "enum": [
                    "inverse-color"
                ]
            },
            "propertyOrder": 20
        },
        "title": {
            "title": "Titulo",
            "description": "Dejar en blanco
para titulo por defecto de metadatos. ",
            "type": "string",
            "propertyOrder": 60,
            "options": {
                "grid_columns": 6
            }
        },
        "description": {
            "title": "Descripcion",
            "oneOf": [
                {
                    "type": "object",

```

```

            "title": "Por defecto",
            "properties": {
                "descripcion": {
                    "title": "Descripcion",
                    "type": "string",
                    "options": {
                        "hidden": true
                    }
                },
                "tipo": {
                    "type": "string",
                    "default": "default",
                    "template": "default",
                    "format": "hidden",
                    "options": {
                        "hidden": true
                    }
                }
            },
            "auxdefault": {
                "type": "string",
                "default": "fondo",
                "template": "fondo",
                "format": "hidden",
                "options": {
                    "hidden": true
                }
            }
        },
        {
            "type": "object",
            "title": "Personalizado",
            "properties": {
                "descripcion": {
                    "title": "Descripcion",
                    "type": "string"
                },
                "tipo": {
                    "type": "string",
                    "default": "custom",
                    "template": "custom",
                    "format": "hidden",
                    "options": {
                        "hidden": true
                    }
                },
                "auxcustom": {
                    "type": "string",
                    "default": "fondo",
                    "template": "fondo",
                    "format": "hidden",
                    "options": {
                        "hidden": true
                    }
                }
            }
        },
        {
            "type": "object",
            "title": "Vacio",
            "properties": {
                "descripcion": {
                    "title": "Descripcion",
                    "type": "string",
                    "options": {
                        "hidden": true
                    }
                },
                "tipo": {
                    "type": "string",
                    "default": "empty",
                    "template": "empty",
                    "format": "hidden",
                    "options": {
                        "hidden": true
                    }
                },
                "auxempty": {
                    "type": "string",

```

```

        "default": "fondo",
        "template": "fondo",
        "format": "hidden",
        "options": {
            "hidden": true
        }
    }
}
},
"options": {
    "collapsed": false,
    "disable_collapse": true,
    "disable_edit_json": true,
    "disable_properties": true,
    "grid_columns": 6
},
},
"backgrounds": {
    "type": "array",
    "format": "table",
    "title": "Fondos",
    "uniqueItems": true,
    "items": {
        "oneOf": [
            {
                "type": "object",
                "title": "Imagen de fondo",
                "headerTemplate": "Imagen
fondo",
                "properties": {
                    "src": {
                        "title": "Fondo",
                        "type": "string",
                        "format": "url",
                        "options": {
                            "upload": true
                        },
                        "links": [
                            {
                                "href":
"../{{self}}",
                                "": "ver imagen"
                            }
                        ]
                    },
                    "position": {
                        "title": "Background
position",
                        "type": "string",
                        "enum": [
                            "left top",
                            "left center",
                            "left bottom",
                            "right top",
                            "right center",
                            "right bottom",
                            "center top",
                            "center center",
                            "center bottom"
                        ],
                        "default": "center
bottom"
                    },
                    "tipo": {
                        "type": "string",
                        "default": "fondo",
                        "template": "fondo",
                        "format": "hidden",
                        "options": {
                            "hidden": true
                        }
                    },
                    "auxfondo": {
                        "type": "string",
                        "default": "fondo",
                        "template": "fondo",
                        "format": "hidden",

```

```

        "options": {
            "hidden": true
        }
    }
},
{
    "type": "object",
    "title": "Imagen de fondo
por defecto",
    "headerTemplate": "Imagen de
fondo por defecto de la pagina",
    "properties": {
        "position": {
            "title": "Background
position",
            "type": "string",
            "enum": [
                "left top",
                "left center",
                "left bottom",
                "right top",
                "right center",
                "right bottom",
                "center top",
                "center center",
                "center bottom"
            ],
            "default": "center
bottom"
        },
        "tipo": {
            "type": "string",
            "default":
"fondodefecto",
            "template":
"fondodefecto",
            "format": "hidden",
            "options": {
                "hidden": true
            }
        },
        "auxfondodefecto": {
            "type": "string",
            "default":
"fondodefecto",
            "template":
"fondodefecto",
            "format": "hidden",
            "options": {
                "hidden": true
            }
        },
        {
            "type": "object",
            "title": "Elegir patron
repeticion",
            "headerTemplate": "Elegir
patron",
            "properties": {
                "src": {
                    "title": "Elegir
patron",
                    "type": "string",
                    "format": "url",
                    "enum": [],
                    "options": {
                        "enum_titles": [],
                        "help": "Enum y
enum_titles se rellenan
programaticamente del contenido de la
carpeta img/overlays/"
                    },
                    "links": [
                        {
                            "href":
"../{{self}}",

```

```

        "rel": "ver imagen",
        "mediaType":
"image/png"
    }
    },
    "tipo": {
        "type": "string",
        "default":
"elegirpatron",
        "template":
"elegirpatron",
        "format": "hidden",
        "options": {
            "hidden": true
        }
    },
    "auxelegirpatron": {
        "type": "string",
        "default":
"elegirpatron",
        "template":
"elegirpatron",
        "format": "hidden",
        "options": {
            "hidden": true
        }
    }
},
{
    "type": "object",
    "title": "Subir Patron
Repeticion",
    "headerTemplate": "Subir
Patron",
    "properties": {
        "src": {
            "title": "Subir patron",
            "type": "string",
            "format": "url",
            "options": {
                "upload": true
            },
            "links": [
                {
                    "href":
"../{{self}}",
                    "rel": "ver imagen",
                    "mediaType":
"image/png"
                }
            ]
        },
        "tipo": {
            "type": "string",
            "default":
"subirpatron",
            "template":
"subirpatron",
            "format": "hidden",
            "options": {
                "hidden": true
            }
        },
        "auxsubirpatron": {
            "type": "string",
            "default":
"subirpatron",
            "template":
"subirpatron",
            "format": "hidden",
            "options": {
                "hidden": true
            }
        }
    }
},
{
    "type": "object",
    "title": "Subir Tira de
Repeticion",
    "headerTemplate": "Subir
Tira",
    "properties": {
        "src": {
            "title": "Subir tira",
            "type": "string",
            "format": "url",
            "options": {
                "upload": true
            },
            "links": [
                {
                    "href":
"../{{self}}",
                    "rel": "ver imagen"
                }
            ]
        },
        "tipo": {
            "type": "string",
            "default": "subirtira",
            "template": "subirtira",
            "format": "hidden",
            "options": {
                "hidden": true
            }
        },
        "auxsubirtira": {
            "type": "string",
            "default": "subirtira",
            "template": "subirtira",
            "format": "hidden",
            "options": {
                "hidden": true
            }
        }
    },
    "options": {
        "collapsed": false,
        "disable_collapse": true,
        "disable_edit_json": true,
        "disable_properties": true
    }
}
}
}
}

```

```

        "type": "object",
        "title": "Subir Tira de
Repeticion",
        "headerTemplate": "Subir
Tira",
        "properties": {
            "src": {
                "title": "Subir tira",
                "type": "string",
                "format": "url",
                "options": {
                    "upload": true
                },
                "links": [
                    {
                        "href":
"../{{self}}",
                        "rel": "ver imagen"
                    }
                ]
            },
            "tipo": {
                "type": "string",
                "default": "subirtira",
                "template": "subirtira",
                "format": "hidden",
                "options": {
                    "hidden": true
                }
            },
            "auxsubirtira": {
                "type": "string",
                "default": "subirtira",
                "template": "subirtira",
                "format": "hidden",
                "options": {
                    "hidden": true
                }
            }
        },
        "options": {
            "collapsed": false,
            "disable_collapse": true,
            "disable_edit_json": true,
            "disable_properties": true
        }
    }
}
}
}
}

```

9.10 Estructura de Columnas

```

{
    "title": "Estructura de n columnas",
    "type": "object",
    "properties": {
        "template": {
            "title": "Plantilla",
            "type": "string",
            "template": "columnas_simples.tpl"
        },
        "groupTemplate": {
            "title": "Plantilla del grupo",
            "type": "string",
            "template":
"columnas_simples_group.tpl"
        },
        "xs": {
            "title": "Dispositivos moviles",
            "type": "string",
            "enum": [
                "1",
                "2",

```

```

        "3",
        "4",
        "6",
        "12"
    ],
    "propertyOrder": 10
},
"sm": {
    "title": "Tabletas",
    "type": "string",
    "enum": [
        "1",
        "2",
        "3",
        "4",
        "6",
        "12"
    ],
    "propertyOrder": 20
},
"md": {
    "title": "Ordenadores pantalla pequeña",
    "type": "string",
    "enum": [
        "1",
        "2",
        "3",
        "4",
        "6",
        "12"
    ],
    "propertyOrder": 30
},
"lg": {
    "title": "Ordenadores pantalla normal",
    "type": "string",
    "enum": [
        "1",
        "2",
        "3",
        "4",
        "6",
        "12"
    ],
    "propertyOrder": 40
}
}
}

```

9.11 Estructura de Puntos Cardinales

```

{
    "title": "Estructura de puntos cardinales",
    "type": "object",
    "properties": {
        "template": {
            "title": "Plantilla",
            "type": "string",
            "template": "puntos_cardinales.tpl"
        }
    }
}

```

9.12 Estructura de Pestañas

```

{
    "title": "Estructura pestanas",
    "type": "object",
    "properties": {
        "template": {

```

```

            "title": "Plantilla",
            "type": "string",
            "enum": [
                "pestanas.tpl",
                "pestanas_seo.tpl",
                "pestanas_seo_sin_titulo.tpl",
                "pestanas_diga33.tpl"
            ],
            "default": "pestanas.tpl"
        },
        "dynamicGaps": {
            "type": "array",
            "format": "table",
            "title": "Pestanas",
            "uniqueItems": true,
            "items": {
                "type": "object",
                "title": "Pestana",
                "headerTemplate": "{{i}} - {{self.titulo}}",
                "properties": {
                    "titulo": {
                        "type": "string",
                        "title": "Titulo"
                    }
                }
            },
            "gapName": {
                "title": "Nombre Huecos",
                "type": "string",
                "template": "pestana",
                "format": "hidden"
            }
        }
    }
}

```

9.13 Estructura de Diapositivas

```

{
    "title": "Pase de diapositivas",
    "type": "object",
    "properties": {
        "template": {
            "title": "Plantilla",
            "type": "string",
            "template": "owl_carousel_1.tpl"
        },
        "groupTemplate": {
            "title": "Plantilla del grupo",
            "type": "string",
            "template": "owl_carousel_1_group.tpl"
        }
    }
}

```

9.14 Decorador de Título

```

{
    "title": "Decorador título",
    "type": "object",
    "description": "Coloca un titulo en la parte superior del modulo decorado. ",
    "properties": {
        "template": {
            "title": "Plantilla",
            "type": "string",
            "enum": [
                "decorador_titulo.tpl",
                "decorador_titulo_alt.tpl",
                "decorador_titulo_gris.tpl",
                "decorador_titulo_gris_subrayado.tpl"
            ],
            "propertyOrder": 1
        },

```

```

"titulo": {
  "title": "Título",
  "type": "string",
  "propertyOrder": 10
},
"nivel_titulo": {
  "title": "Nivel titulo",
  "type": "string",
  "enum": [
    "h1",
    "h2",
    "h3",
    "h4"
  ],
  "default": "h2",
  "propertyOrder": 20
}
}

```

9.15 Decorador de Enlace

```

{
  "title": "Decorador de enlace",
  "description": "No utilizar NUNCA cuando existan enlaces dentro. ",
  "type": "object",
  "properties": {
    "template": {
      "title": "Plantilla",
      "type": "string",
      "enum": [
        "decorador_enlace.tpl"
      ],
      "default": "decorador_enlace.tpl",
      "propertyOrder": 10
    },
    "href": {
      "title": "Enlace",
      "type": "string",
      "default": "https://",
      "propertyOrder": 20
    },
    "title": {
      "title": "Título",
      "type": "string",
      "default": "",
      "propertyOrder": 30
    }
  }
}

```

9.16 Decorador de Bloque

```

{
  "title": "Decorador bloque",
  "type": "object",
  "description": "Coloca este decorador el ultimo para que incluya al resto de decoradores y queden dentro. ",
  "properties": {
    "template": {
      "title": "Plantilla",

```

```

      "type": "string",
      "template": "decorador_bloque.tpl",
      "propertyOrder": 10
    },
    "bg_color": {
      "title": "Color",
      "type": "string",
      "enum": [
        "transparent",
        "primary",
        "white",
        "info",
        "success",
        "warning",
        "danger"
      ],
      "propertyOrder": 20
    },
    "padding": {
      "title": "Margen interior (padding)",
      "type": "string",
      "enum": [
        "no",
        "todos_lados",
        "superior_inferior",
        "izquierda_derecha"
      ],
      "propertyOrder": 30
    },
    "margin": {
      "title": "Margen exterior (margin)",
      "type": "string",
      "enum": [
        "no",
        "superior",
        "inferior",
        "superior_inferior"
      ],
      "propertyOrder": 40
    },
    "borde": {
      "title": "Borde",
      "type": "string",
      "enum": [
        "sin_borde"
      ],
      "propertyOrder": 50
    },
    "class": {
      "title": "Estra classes",
      "type": "string",
      "propertyOrder": 60
    },
    "style": {
      "title": "Style",
      "type": "string",
      "propertyOrder": 70
    }
  }
}

```


10 Anexo II: JSON Schema de las páginas de CMS

```
{
  "title": "cms_page",
  "type": "object",
  "properties": {
    "cms_pages_title": {
      "type": "string",
      "options": {
        "grid_columns": 12
      },
      "propertyOrder": 10
    },
    "cms_pages_description": {
      "title": "descripcion",
      "oneOf": [
        {
          "title": "code",
          "type": "string",
          "format": "html",
          "options": {
            "wysiwyg": false
          }
        },
        {
          "title": "wysiwyg",
          "type": "string",
          "format": "html",
          "options": {
            "wysiwyg": true
          }
        }
      ],
      "options": {
        "grid_columns": 12
      },
      "propertyOrder": 20
    },
    "cms_pages_status": {
      "type": "integer",
      "enum": [
        0,
        1
      ],
      "options": {
        "grid_columns": 3,
        "enum_titles": [
          "Desactivado",
          "Activado"
        ]
      },
      "propertyOrder": 30
    },
    "cms_pages_order": {
      "type": "integer",
      "options": {
        "grid_columns": 3
      },
      "propertyOrder": 40
    },
    "cms_pages_from": {
      "title": "Fecha desde (AAAA-MM-DD HH:MM:SS)",
      "type": "string",
      "format": "datetime",
      "options": {
        "grid_columns": 6
      },
      "propertyOrder": 50
    },
    "cms_pages_to": {
      "title": "Fecha hasta (AAAA-MM-DD HH:MM:SS)",
      "type": "string",
      "format": "datetime",
      "options": {
        "grid_columns": 6
      },
    },
  },
}
```

```
    "propertyOrder": 60
  },
  "cms_pages_url": {
    "type": "string",
    "options": {
      "grid_columns": 6
    },
    "propertyOrder": 70
  },
  "cms_pages_sitemap_priority": {
    "type": "integer",
    "options": {
      "grid_columns": 3
    },
    "propertyOrder": 40
  },
  "cms_pages_menu_name": {
    "type": "string",
    "options": {
      "grid_columns": 6
    },
    "propertyOrder": 70
  },
  "cms_pages_template_master_name": {
    "type": "string",
    "enum": [
      'implode(',',',
      array_merge(array(''), $modPagesDir)
    ],
    "options": {
      "grid_columns": 6
    },
    "propertyOrder": 80
  },
  "cms_pages_template_master_id": {
    "type": "string",
    "options": {
      "grid_columns": 6
    },
    "propertyOrder": 90
  },
  "cms_pages_keywords": {
    "type": "string",
    "options": {
      "grid_columns": 6
    },
    "propertyOrder": 100
  },
  "cms_pages_image": {
    "type": "string",
    "format": "url",
    "options": {
      "upload": true,
      "grid_columns": 6
    },
    "links": [
      {
        "href": "../{{self}}",
        "rel": "ver imagen"
      }
    ],
    "propertyOrder": 110
  },
  "options": {
    "collapsed": false,
    "disable_collapse": true,
    "disable_edit_json": true,
    "disable_properties": true
  }
}
```